MicroStar International
MR-W1
Intelligent Indoor Robotic Toolkits

機器人載具模組操作手冊

Version 0C



Robot Team

Research Center @ MSI

Apr 3, 2008

安全性說明

- ✓ 在使用 MR-W1 機器人之前，請先閱讀操作手冊。
- ✓ 不可將 MR-W1 機器人淋雨或放置在潮濕處，否則有觸電的危險。
- ✓ 不可自行開啟 MR-W1 機器人內部元件或配件。
- ✓ 不要將輪子駛過長髮、長毛、細線處。
- ✓ 不可在充電狀況或是電源開啟時打開 MR-W1 機器人。

不適當操作狀況

若不適當地操作 MR-W1 機器人，會喪失保固的資格！不適當操作包含，但不限於以下狀況：

- ✓ 摔落 MR-W1 機器人、撞擊突出或尖銳處、或是操控 MR-W1 機器人任意亂撞。
- ✓ 將 MR-W1 機器人承載超過負載上限的重物。
- ✓ 將 MR-W1 機器人淋溼。
- ✓ 重複地將 MR-W1 機器人駛過長髮、長毛、細線處，或其他會使輪子被纏繞的物品。
- ✓ 在充電狀況或是電源開啟時打開 MR-W1 機器人。

目錄

第一章 概述......5
基本元件......5
選購元件......5
系統需求......5
額外資源......5
技術支援......6
第二章 MR-W1 機器人介紹......7
硬體特性......7
軟體特性......7
第三章 MR-W1 機器人規格......8
硬體特性......8
指針型電壓表......8
超音波感測器......8
紅外線感測器......9
馬達、輪子與編碼器......10
電池及電力系統......10
第四章 快速入門......11
硬體組裝......11
安裝電池......11
與筆記型電腦連接......13
測試程式......16

第一章 概述

感謝您購買 MR-W1 機器人！在您使用或發展機器人軟硬體技術之前，請先閱讀以下的技術細節。

基本元件

- ✓ MR-W1 機器人本體(含電池)
- ✓ 含 MR-W1 軟體及說明文件的光碟
- ✓ 保險絲
- ✓ USB 轉接線

選購元件

- ✓ 電池充電器
- ✓ 筆記型電腦
- ✓ 備用電池
- ✓ Pan-tilt 影像系統(配合影像軟體須選購)
- ✓ 全像式影像系統

系統需求

- ✓ 586 或以上的 PC
- ✓ Windows XP 或以上的作業系統
- ✓ USB 連接埠
- ✓ 100M 以上的硬碟

額外資源

任何購買 MR-W1 機器人的新客戶都能享有以下額外資源：

- ✓ 下載或一年免費更新最新版軟體、韌體、以及說明文件
- ✓ MR-W1 機器人技術支援部門的直接服務

技術支援

若您在使用 MR-W1 機器人上有任何問題，而說明文件中找不到解答時，請直接寫電子郵件到

octoberhung@msi.com.tw
02-32345599#1039

電子郵件請附上您購買的 MR-W1 機器人序號，以便我們能在最短的時間解答您的問題。

第二章 MR-W1 機器人介紹

MR-W1 機器人平台是由堅實的鋁製外殼、兩個主動輪、兩個輔輪、直流馬達、控制系統、驅動系統、高解析度解碼器、電力系統、及機器人軟體控制系統所組成。



Figure 2.1 MR-W1 robot

硬體特性

- ✓ 150 MHz 32-bit, fixed-point C2812 DSP core (含 32K RAM and 128k FLASH)
- ✓ 標準 6 個，最多支援 23 個超音波感測器讀取
- ✓ 最多支援 15 個短紅外線感測器，標準支援 11 個短紅外線感測器
- ✓ 最多支援 15 個短紅外線感測器，標準支援 4 個長紅外線感測器
- ✓ 指針式電壓表

軟體特性

使用 MR-W1 機器人的控制板及控制系統，可以完成低階的機器人控制，包含位置、速度、加速度控制、感測器資訊讀取、機器人狀態偵測、電壓偵測等。透過和控制板相連的筆記型電腦，可以編寫機器人軟體控制系統，以完成高階智慧型的機器人控制，包含避障、路徑規畫、定位系統、導航等。

第三章 MR-W1 機器人規格

硬體特性

MR-W1 機器人總重 17kg (包含兩顆電池，一顆約 2.6 公斤)，雖然重量輕，但其堅實的鋁製外殼及架構能提供非常好的高達 25 公斤的額外承載重量能力。

MR-W1 機器人包含以下組件：

- ✓ 指針型電壓表
- ✓ 超音波感測器
- ✓ 紅外線感測器
- ✓ 馬達、輪子與編碼器
- ✓ 電池與電力系統

指針型電壓表

透過指針型電壓表，您可以輕易地偵測 MR-W1 機器人的電源狀況。若是電源低於 22 伏特，請您要盡快將電池充電。

超音波感測器

MR-W1 機器人共裝置了 23 顆超音波感測器[1]，可用於障礙物偵測、避障、特徵識別、定位、以及導航等功能。每兩顆超音波間隔 18 度。提供 1~50 公分的偵測距離。其中底層配置了 12 顆超音波(有 1 顆放置在後方)，而上層配置了 11 顆超音波。

[1] 視購買版本而定，機器人配置超音波可能會有 6~23 顆不等的數量差異。

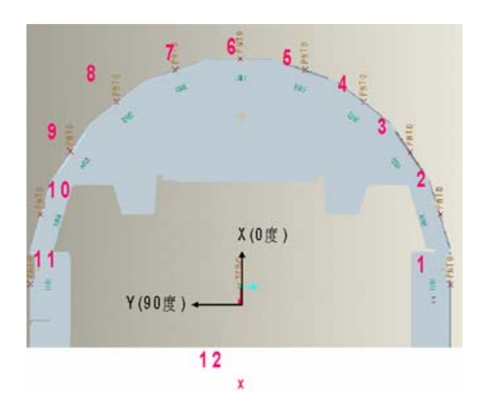


圖 3.1：底層超音波配置

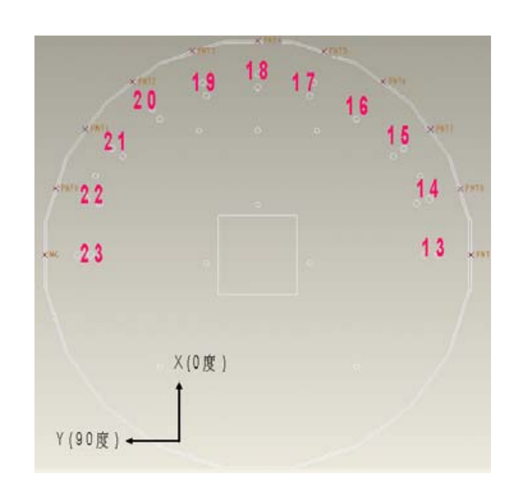


圖 3.2：上層超音波配置

紅外線感測器

MR-W1 機器人共裝置了 15 顆紅外線感測器，同樣地也可用於障礙物偵測、避障、特徵識別、定位、以及導航等功能。每兩顆超音波間隔 18 度。其中有 11 顆為短紅外線感測器，提供 4~25 公分的偵測距離；另外 4 顆為長紅外線感測器，提供 20~100 公分的偵測距離。紅外線感測器同樣也是間隔 18 度來放置。所有的紅外線感測器都放置在底層。

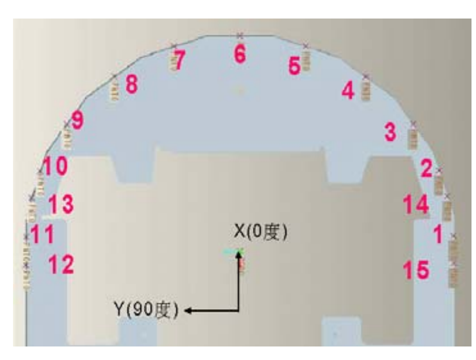


圖 3.2：底層紅外線配置

馬達、輪子與編碼器

MR-W1 機器人的驅動系統使用高速、高扭力的直流馬達，並配備高解析度的光學編碼器，能夠提供精準的位置及速度偵測，以及軌跡估測。MR-W1 機器人使用高規格的非充氣的高彈性輪，不易因承載重量產生形變，且與地面接觸面積小，更能提升軌跡估測的準確度。

電池及電力系統

MR-W1 機器人使用兩個 12 伏特鉛酸電池(7Ahr)。電池壽命會隨著操作情況不同而變化。一般來說，在電池飽和情況下，可以提供五小時的連續使用。

第四章 快速入門

硬體組裝

每個 MR-W1 機器人在出廠前都會經過完整的組裝及測試流程。若您使用自備的筆記型電腦，建議您在剛拿到新的 MR-W1 機器人時，先執行以下的測試，以確保所有的 MR-W1 機器人性能正常。

安裝電池

以下為安裝電池動作。抽出電池動作也相同。若您發現電池電力不足(小於 22 伏特)，要更換電池時，請依照下列步驟來更換電池。

Step 1 打開前蓋扣環



Step 2 將電池滑入電池槽內



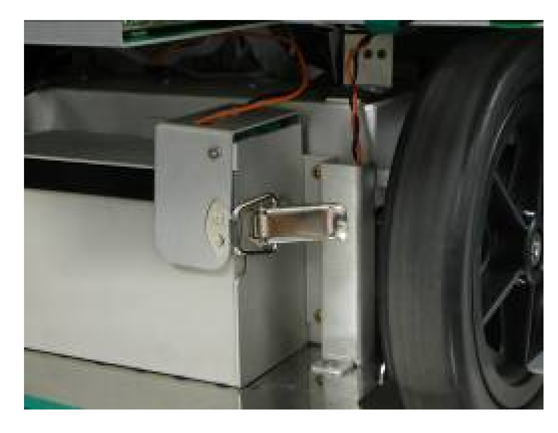
Step 3 扣上電池蓋



Step 4 將電池接點合上



Step 5 將固定環扣上





與筆記型電腦連接

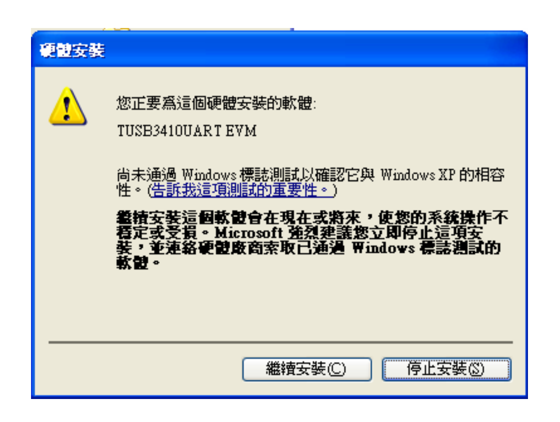
將控制板與筆記型電腦透過 USB 線相連，並將 MR-W1 機器人電源打開。



之後會在 Windows 上出現以下的訊息，請選擇”從清單或特定位置安裝”，將目錄指向光碟上的”TUSB3410_2KXP_V103”資料夾



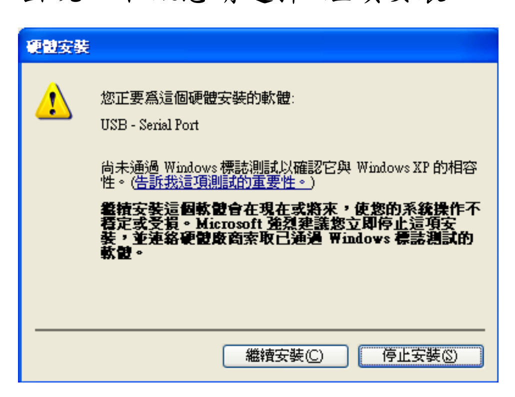
選擇完資料夾後出現以下訊息，並選擇”繼續安裝”。



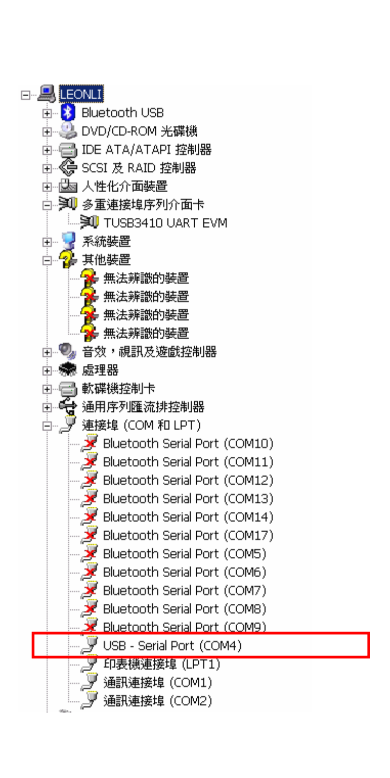
出現以下訊息請選”完成”



出現以下訊息請選擇”繼續安裝”



在 控制台->系統->硬體->裝置管理員 會看到以下資料 請記錄下 USB – Serial Port 的 COM Port，以下圖為例圖示出現的 COM 為 COM4



將光碟中的 MR_W1_Demo 資料夾拷貝到你的電腦硬碟裡
以記事本打開 MR_W1_Demo\conf\Comm.conf 檔將 CommPort 設成在你 PC 上所顯示的 COM Port

Name:Comm .... Object name
Number:1 .... Object quantity

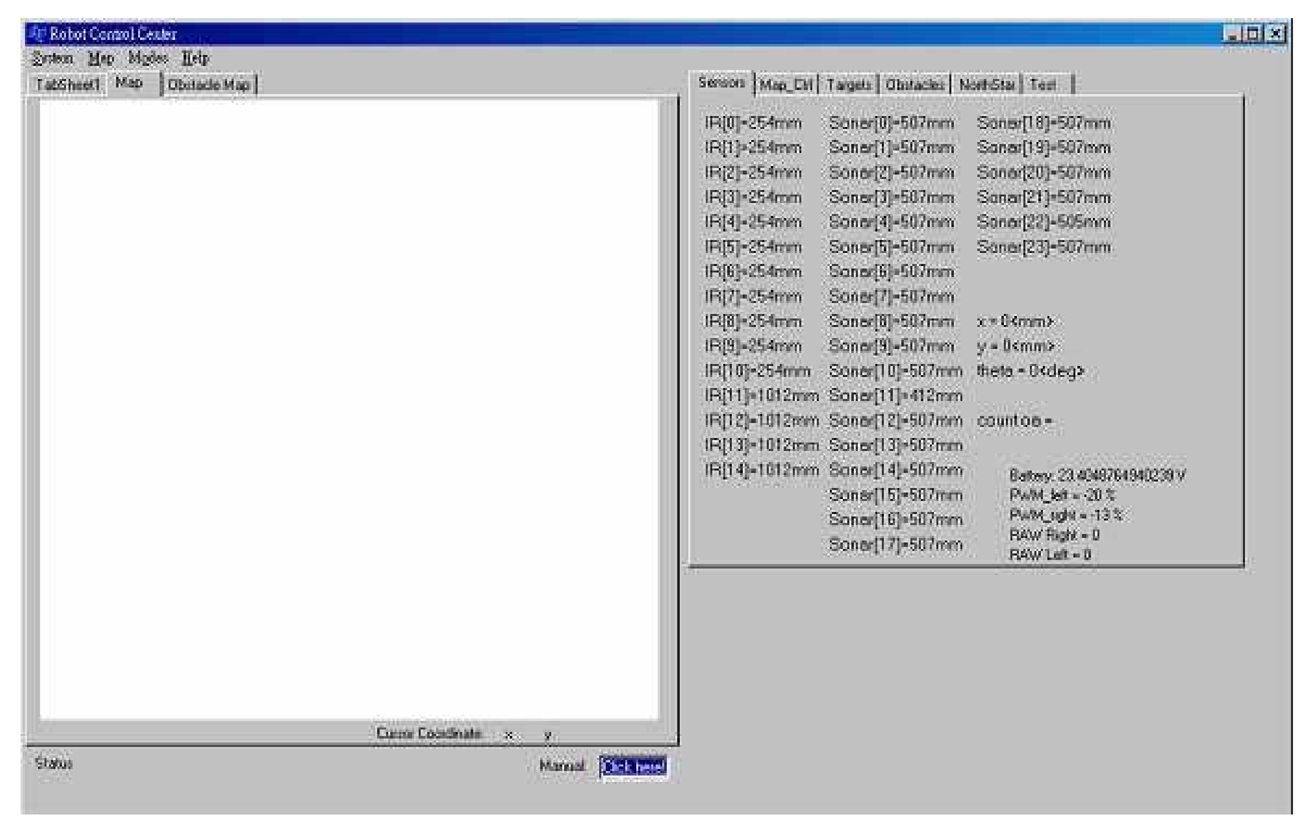
Count:1... Object number
CommPort:COM4... COM port
BaudRate:115200 .... Baud rate
StopBit:0 .... Stop bit
ByteSize:3 .... Data size

//
//CommPort
//COM1~COM10.....
//
//BaudRate
//57600
//115200
//...
//
//StopBit

//0　SB1
//1　SB1.5
//2　SB2
//
//ByteSize
//0　DB5
//1　DB6
//2　DB7
//3　DB8

測試程式

執行 Demo\ Demo.exe 程式您會先看到以下畫面[2]：



1. 首先確定電池電壓是不是有數值。正常的電池電壓範圍在 22V~28V 間。
2. 檢查 IR[0]~IR[14]; Sonar[0]~Sonar[23] 的數值是否正常，隨意拿一個物體在 Sensor 前後移動，確定數值會隨物體前後移動而有所改變。
3. 將滑鼠移到 Click here 並點一下，機器人就可以用方向鍵往前往後移動，或是順、逆時針旋轉。

若 1 至 3 項測試都沒有問題，恭喜您的 MR-W1 機器人是正常運作中。

[2] Demo 程式會持續更新，因此您所拿到的版本可能會和本章中的圖片有些許差異。

有關 Robot Control Center 的詳細使用方法，請詳閱 Demo 程式操作手冊。

MR-W1 校正流程

Version 0A

Robot Team

Research Center

Micro-Star Internaltional

Apr. 03, 2008

目錄

1. 目的..........3

2. 校正流程..........3

1. 目的

目的一：機器人在出貨前，透過完整確實的校正步驟，可讓機器人的控制器在利用馬達 encoder 計算移動距離與角度時，將馬達 encoder 累積誤差值降到最低，確保機器人行走軌跡與實際環境相符合。

目的二：機器人在使用一段時間後，可能因爲碰撞或輪胎損耗等因素，導致移動軌跡不準，此時必須重新校正。或是當機器人確定會在固定環境中執行任務時，可以特別針對該環境將參數作最佳化。也就是說，藉由以下校正流程，可方便使用者自行校正。

2. 校正流程

步驟<一>：粗估兩輪的圓周長

1. 先手動量測左輪直徑(如利用皮尺)
2. 利用(圓周長 = 直徑*pi)公式計算左輪圓周長
   PS : pi = 3.1415926
3. 同理，接著手動測量右輪直徑
4. 透過上述公式計算右輪圓周長

步驟<二>：粗估兩輪到中心點距離

1. 手動計算兩輪間的輪軸距離 D
2. 假設旋轉中心點位於兩輪之間，因此將此距離 D 除以二，所得距離即爲兩輪到旋轉中心點的初始參考距離。

步驟<三>：更改 **Drive.conf** 檔，**Key in** 參數

1. 運用 WordPad 開啓 config 資料夾中的 Drive.conf 檔(強烈建議先將該檔案備份後再修改)
2. 更改左輪參數，Count:Left 以下的資料爲左輪的參數數據
3. 將左輪圓周長 Key in
   Ex：TireCircumference:615.6→代表左輪圓周長爲 615.6
4. 將左輪到中心的距離 Key in
   Ex：_y:166.9→代表左輪到中心點距離爲 166.9
5. 同上面步驟更改右輪參數(也就是在 Count:Right 以下的資料)
   注意：右輪到中心點的距離爲負值。Ex：_y:-166.9

步驟<四>：實際運用軟體控制機器人校正

首先我們必須將機器人放置於平坦地面，如果地面不平坦，校正結果將不正確。

1、 走直線校正

Step1、在地面畫一條直線。

Step2、控制機器人於直線上行走 5m。

Step3、觀察機器人是否有偏離直線。

Step4、如果機器人偏向右邊，代表右輪圓周長參數值太大，必須開啓 Drive.conf 檔，並更改右輪圓周長，改小後再重新執行 Step1~3。看看修改後的效果是否有改善。反之，若機器人偏向左邊，代表左輪圓周長參數值太大，需改小後再重新執行 Step1~3。

Step5、等到機器人能夠完全不偏離直線，就可以進行距離精度校正。

2、 距離精度校正

Step1、先用軟體設定機器人行走距離 5m。

Step2、行走結束後，測量機器人實際行走的距離看是否爲 5m。

Step3、如果不符合，則必須等比例放大或縮小兩輪的圓周長參數，再重新執行 Step1~2，直到實際行走距離爲 5m。若是實際機器人走不到 5m，就要等比例放大兩輪的圓周長參數；反之，若是實際機器人走超過 5m，就要等比例縮小兩輪的圓周長參數。

Step4、等到機器人的實際移動距離剛好等於 5m 時，就可以進行 theta 角校正。

3、 theta 角校正

Step1、首先設立一個基準點，可預先在地上畫一條線。

Step2、開啓軟體控制機器人對準此基準點，遙控機器人順時針轉 10 圈，也就是再度剛好對準此基準點時，觀察 theta 數值是否爲零度。理論上 theta 數值必須爲零，代表沒有旋轉誤差。

Step3、如果發生有誤差的情況則必須調整左右輪與旋轉中心之比例值，直到 theta 角小於正負 3 度。

Step4、然後逆時針旋轉 10 圈，如果發生誤差則必須調整左右輪與旋轉中心之比例值，直到 theta 角小於正負 3 度。

Step5、若是順時針和逆時針旋轉 10 圈後，theta 角都能小於正負 3 度，代表校正完成。若是對於旋轉角度的精度需求較大時，可以改變要求 theta 角需小於 3 度；反之亦然。

3. 結論

精準調校這些參數後的機器人，在控制器的控制下，可漂亮的走出直線、正確的走出想要的距離以及準確走到在地圖上所點選的目標位置，進而可以實現沿牆行走、避障以及目標點移動等功能。但由於每個環境地形不同，因此所走出來的效果也不盡相同，使用者可依照以上步驟，調整參數，使得機器人符合環境條件，走出理想的結果。

Demo 程式操作手冊

Version 0C

Robot Team

Research Center

May 13, 2008

目錄

1. 目的......3

2. 介面......3

3. 選單......4

4. 訊息、遙控、與展示......36

5. 地圖......43

6. 資訊......45

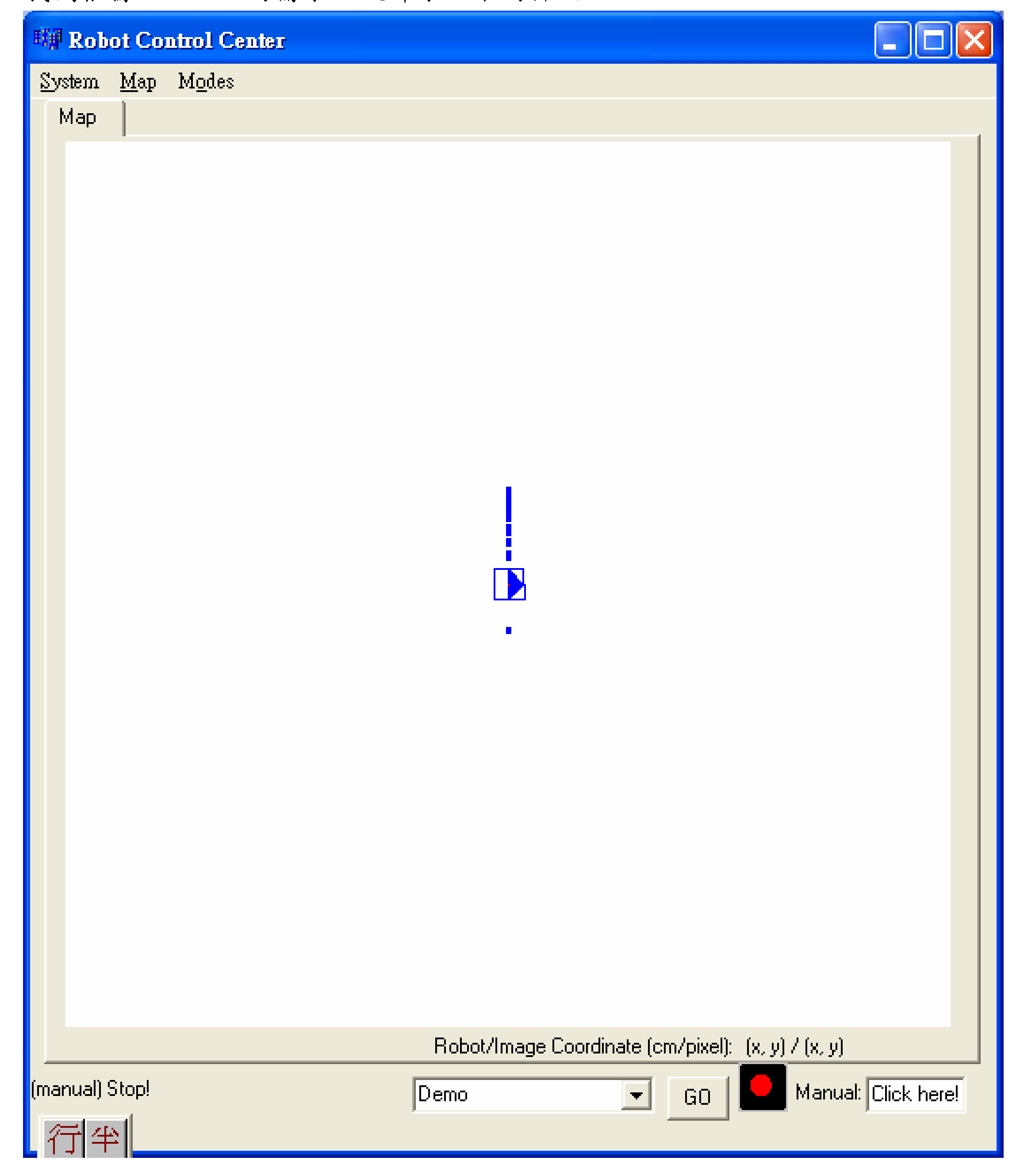
7. 結論......58

1. 目的

本文件介紹 Demo 程式以及如何操作來控制 MR-W1 機器人完成各項功能。

2. 介面

我們根據 MR-W1 的需求，設計了以下的介面：



其中包含四大區塊：

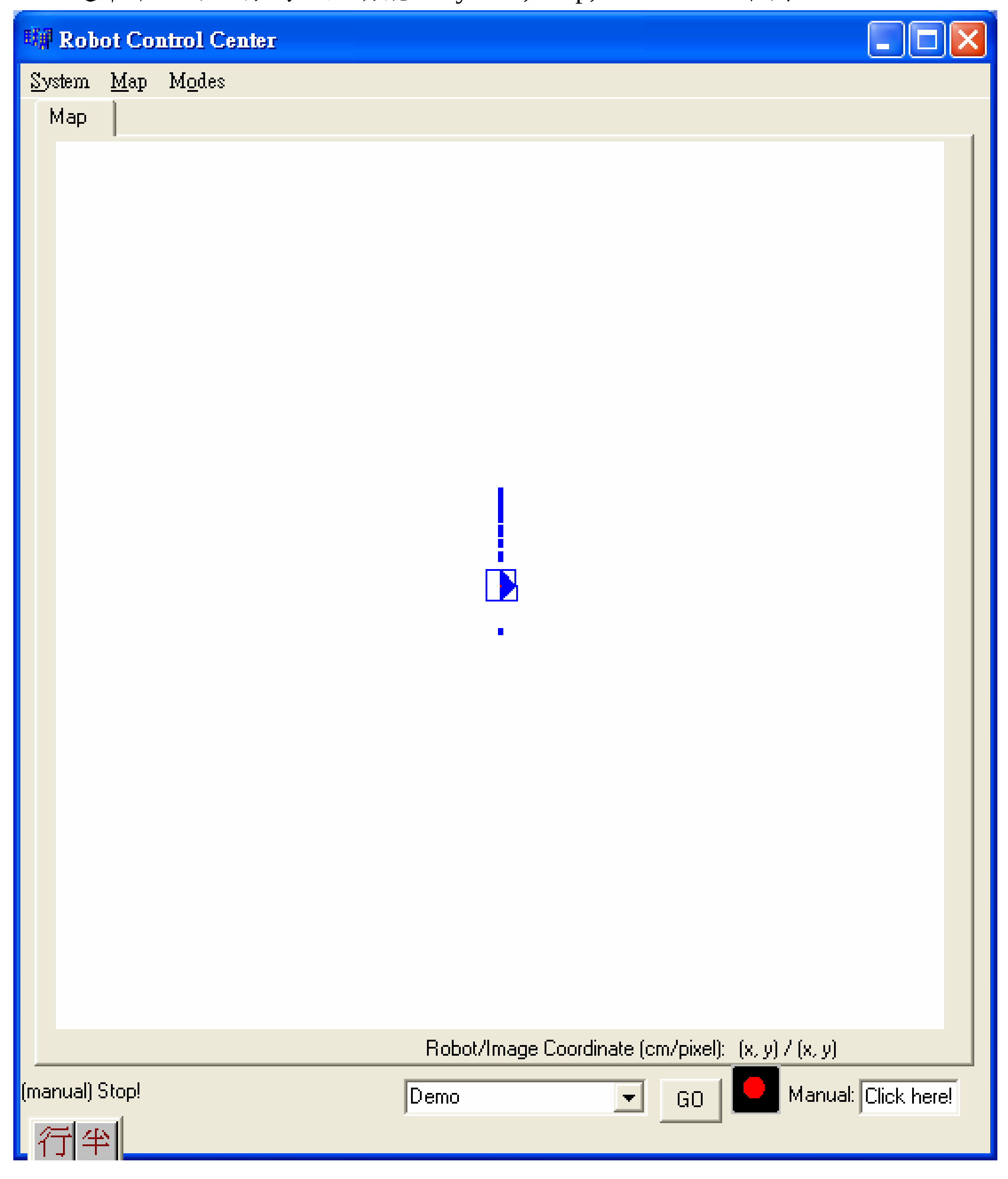
1. 選單區塊

2. 訊息、遙控、與展示區塊
3. 地圖區塊
4. 資訊區塊 (位於 Control Panel 中)

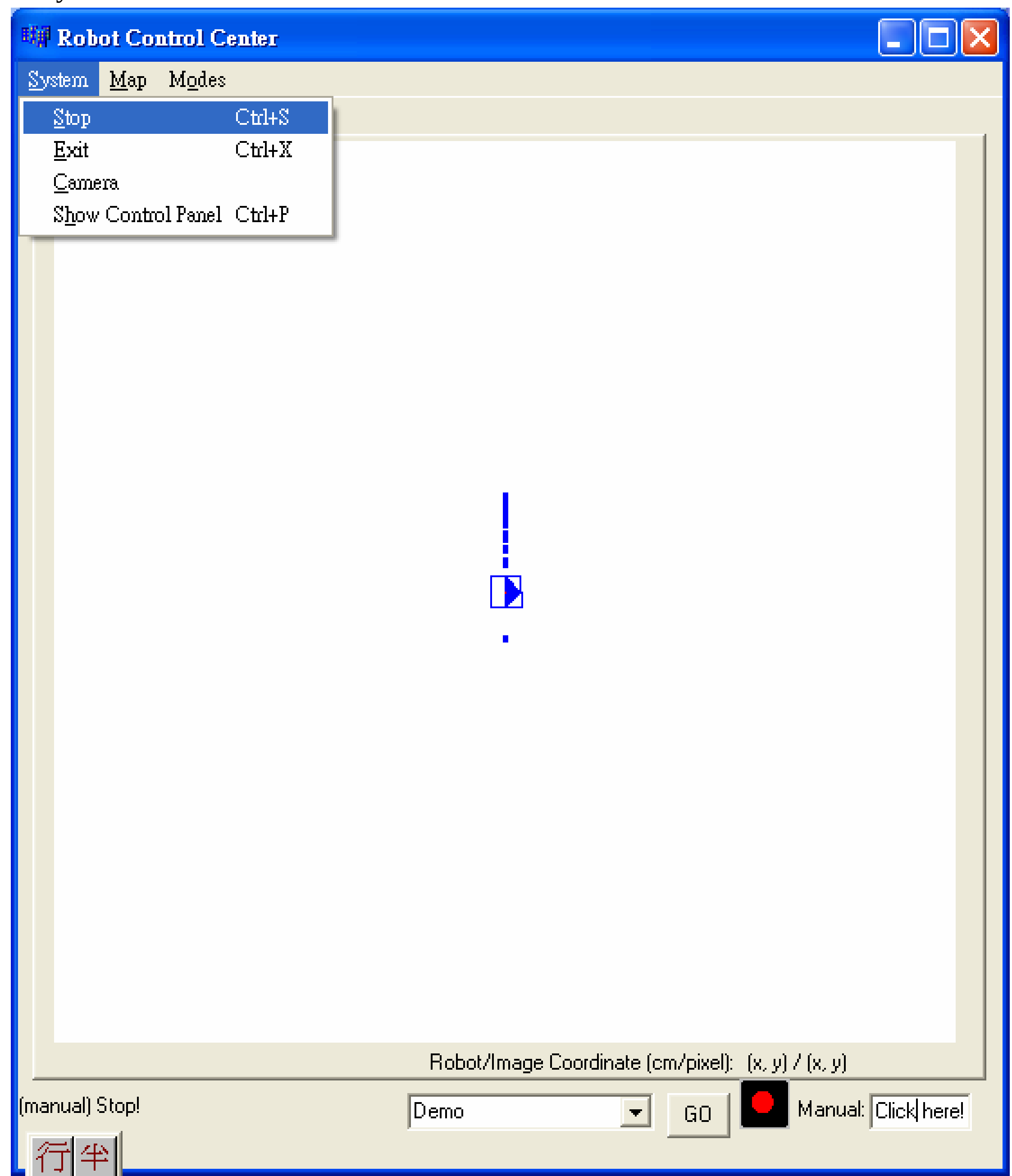
這四大區塊會分別在下面的章節中，做詳細的說明。

3. 選單

選單中又可區分為三大功能：System, Map, Modes，如下圖：



1. System



System 選單中有四個功能：

- Stop (Ctrl+S)

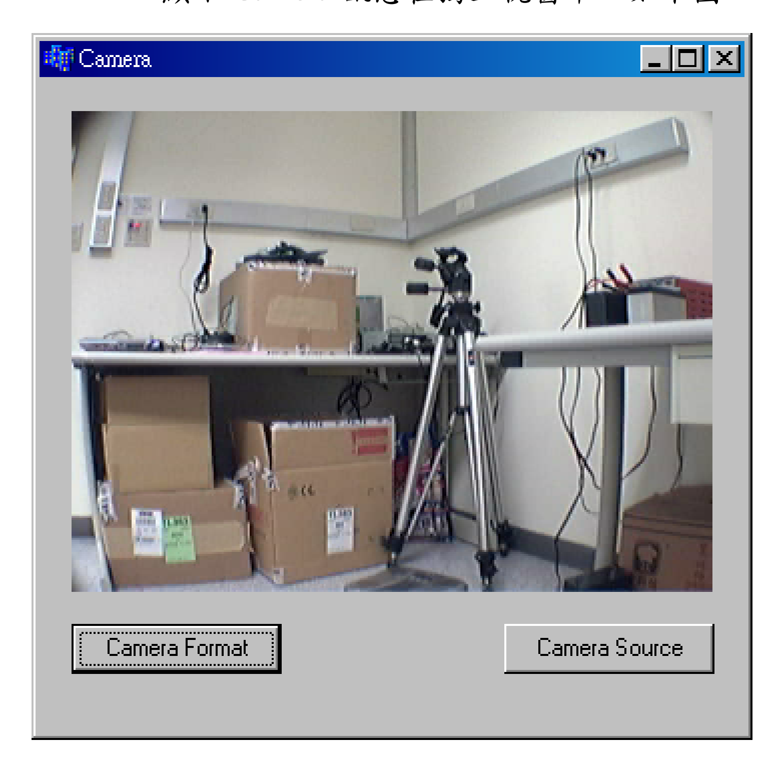
  緊急停止功能。當此功能被按下後，機器人馬達會被鎖死，也就是緊急煞車停住。同時，所有正在執行的任務(mode)都會被取消。

- Exit (Ctrl+X)

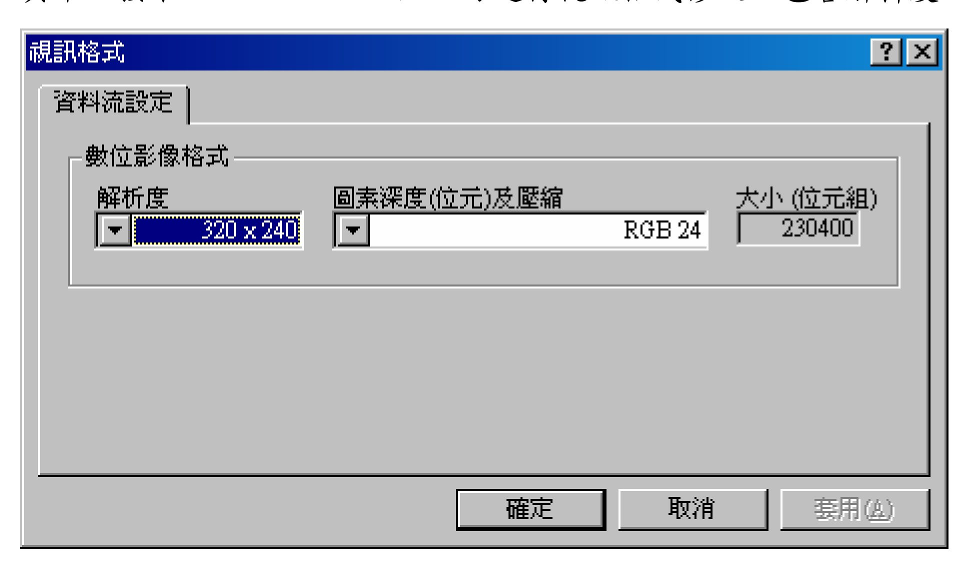
  跳出機器人控制中心程式。

- Camera

顯示 Camera 訊息在獨立視窗中，如下圖：



其中，按下 Camera Format 鈕，可進行視訊格式修改，包含解析度、畫素格式等，如下圖：

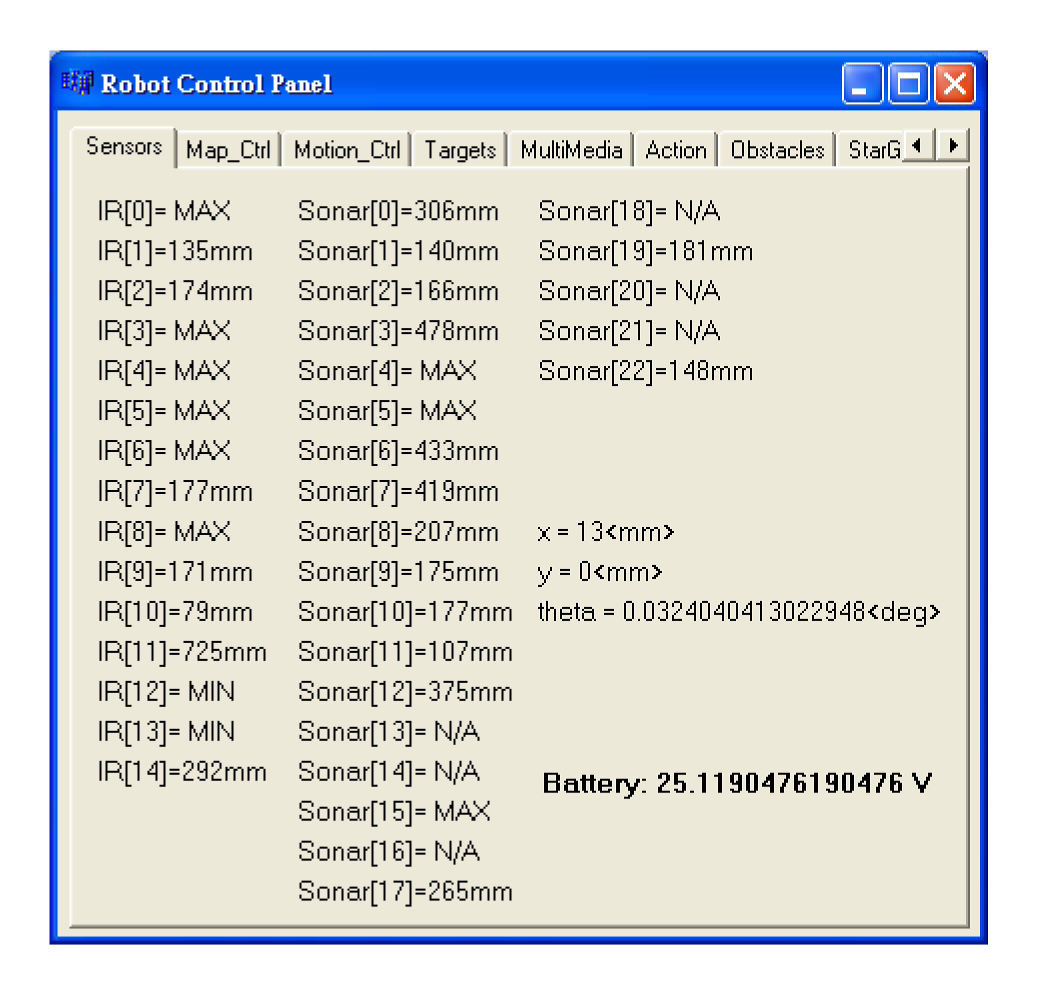


若按下 Camera Source 鈕，可進行視訊來源修改，包含影像控制、影像模式、白平衡設定、閃爍設定等，如下圖：

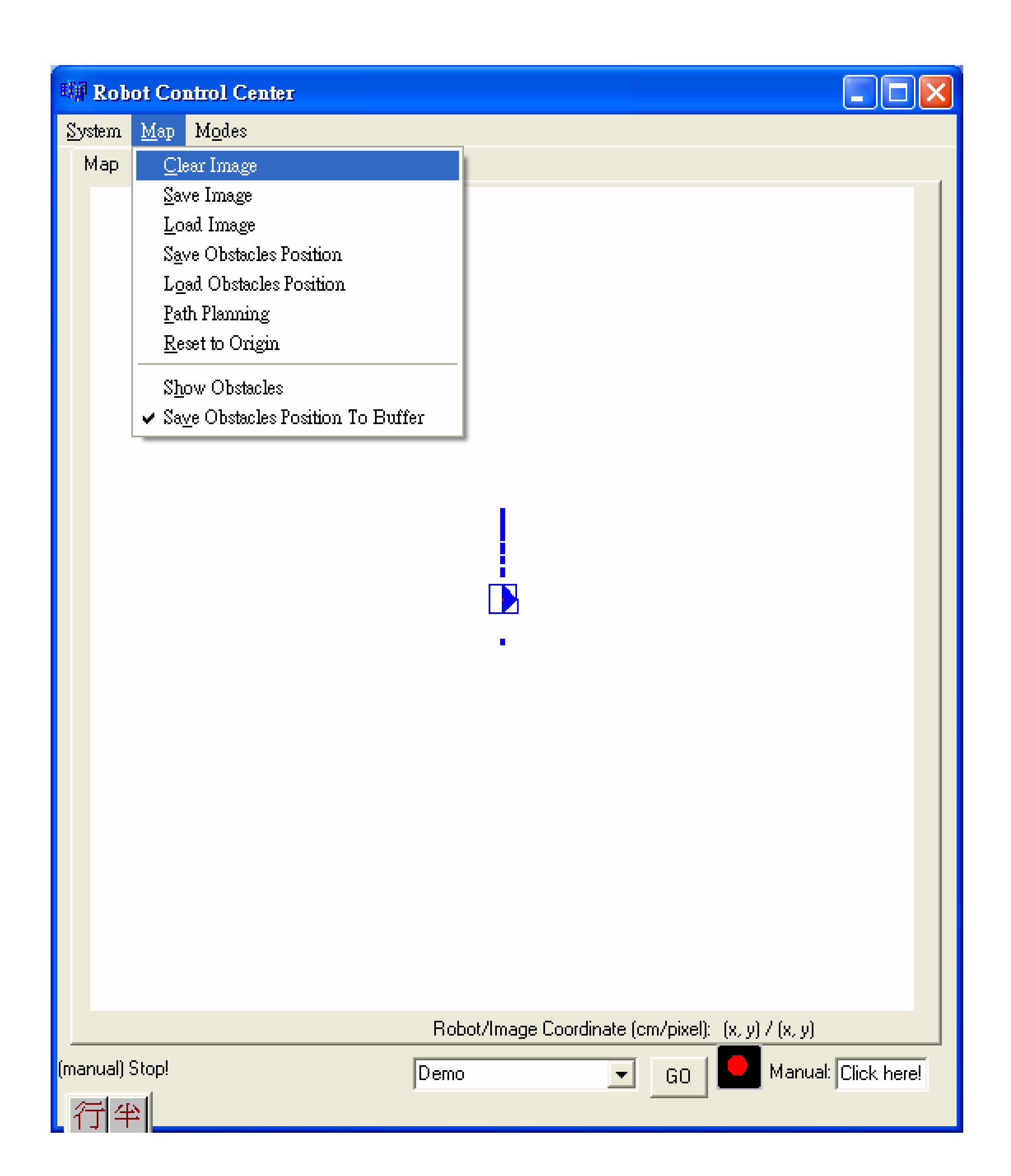


- Show Control Panel (Ctrl+P)

  顯示 Control Panel 在新視窗中，如下圖：


Robot Control Panel
Sensors
Map_Ctrl
Motion_Ctrl
Targets
MultiMedia
Action
Obstacles
StarG
IR[0]= MAX
IR[1]=135mm
IR[2]=174mm
IR[3]= MAX
IR[4]= MAX
IR[5]= MAX
IR[6]= MAX
IR[7]=177mm
IR[8]= MAX
IR[9]=171mm
IR[10]=79mm
IR[11]=725mm
IR[12]= MIN
IR[13]= MIN
IR[14]=292mm
Sonar[0]=306mm
Sonar[1]=140mm
Sonar[2]=166mm
Sonar[3]=478mm
Sonar[4]= MAX
Sonar[5]= MAX
Sonar[6]=433mm
Sonar[7]=419mm
Sonar[8]=207mm
Sonar[9]=175mm
Sonar[10]=177mm
Sonar[11]=107mm
Sonar[12]=375mm
Sonar[13]= N/A
Sonar[14]= N/A
Sonar[15]= MAX
Sonar[16]= N/A
Sonar[17]=265mm
Sonar[18]= N/A
Sonar[19]=181mm
Sonar[20]= N/A
Sonar[21]= N/A
Sonar[22]=148mm
x = 13<mm>
y = 0<mm>
theta = 0.0324040413022948<deg>
Battery: 25.1190476190476 V

2. Map



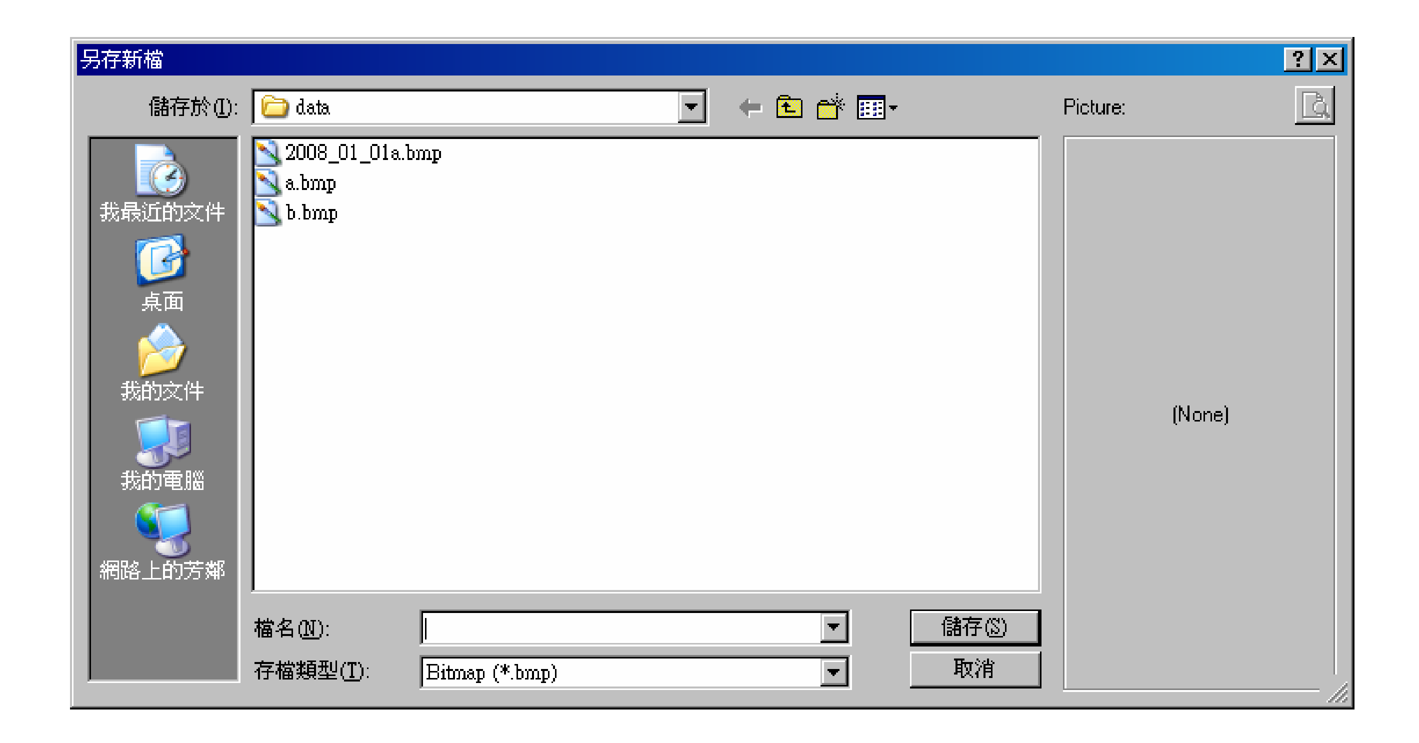
Map 選單中有以下功能：

- Clear Image

  清除地圖區塊內的地圖資訊。

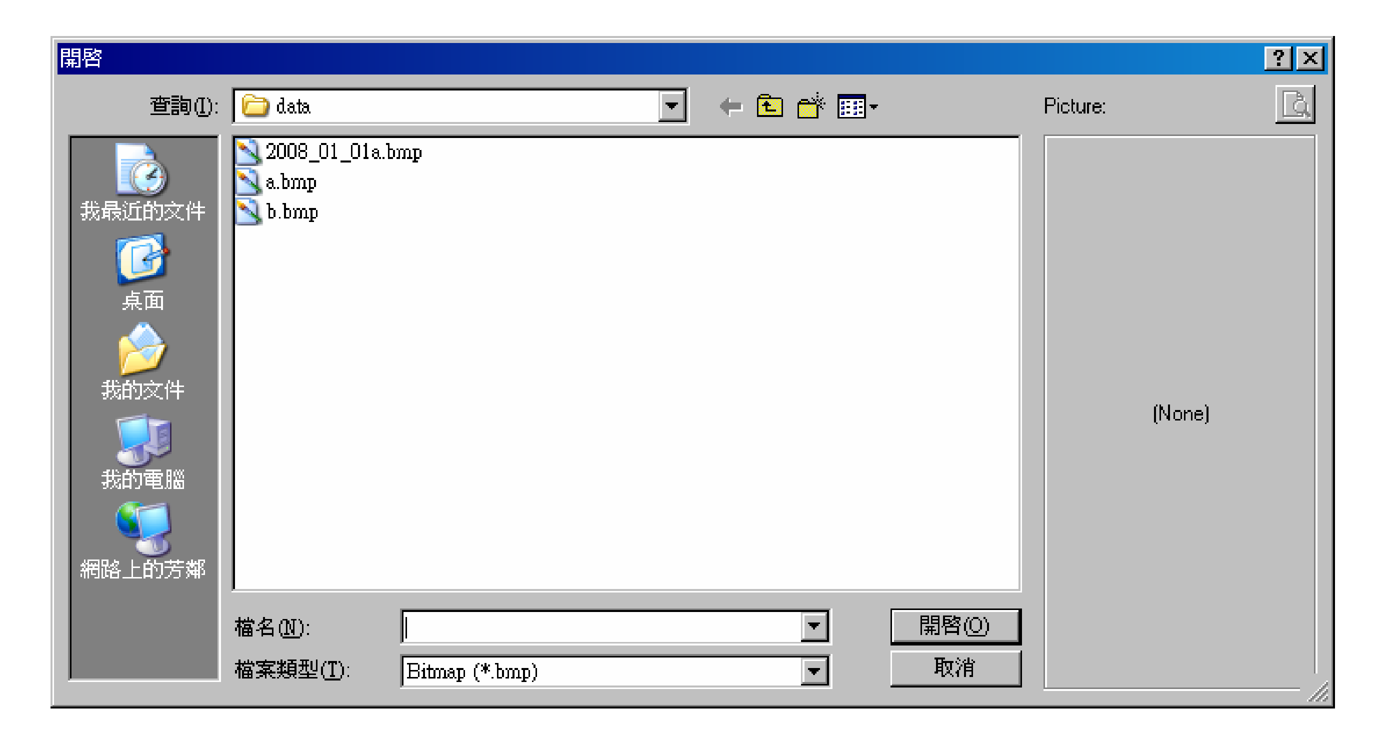
- Save Image

  儲存地圖功能，副檔名為.bmp。能夠將目前地圖區塊的地圖儲存起來，如下圖：



- Load Image

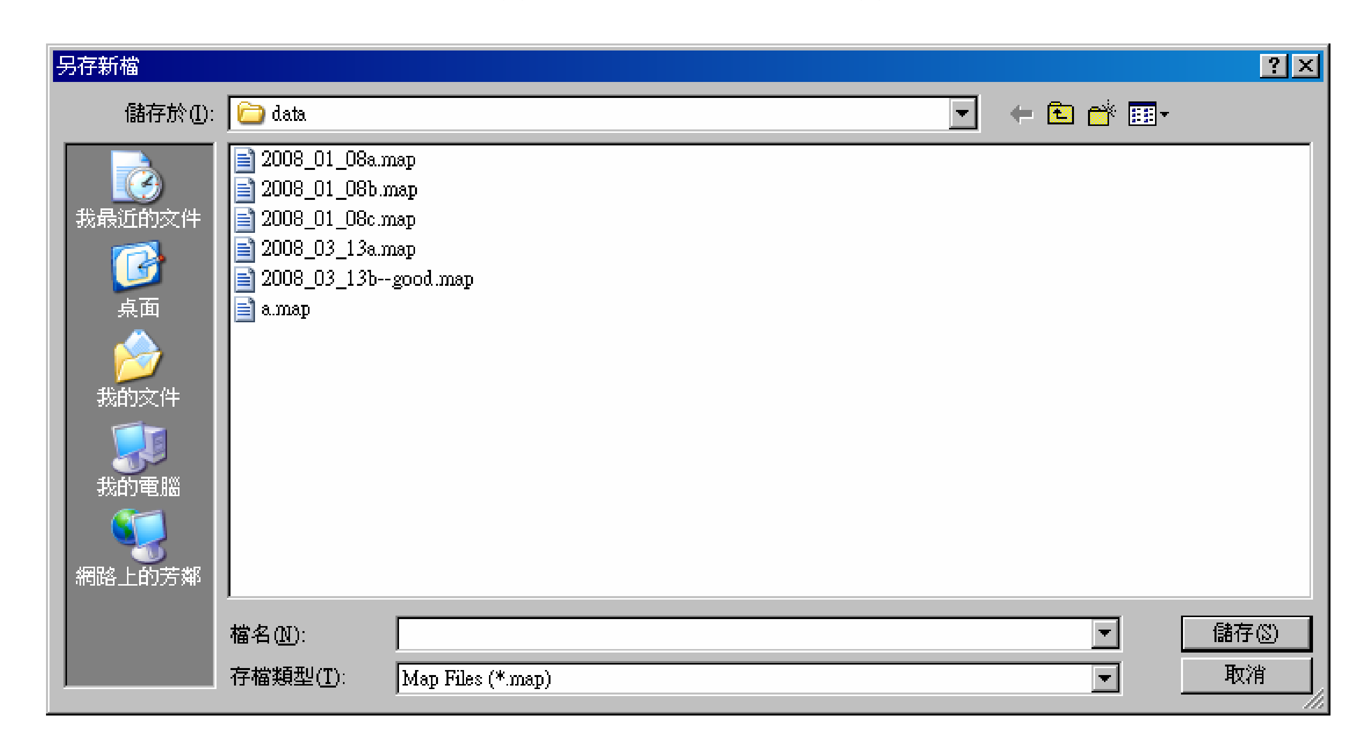
載入地圖功能，副檔名為.bmp。能夠將之前儲存的地圖載入到地圖區塊中，如下圖：

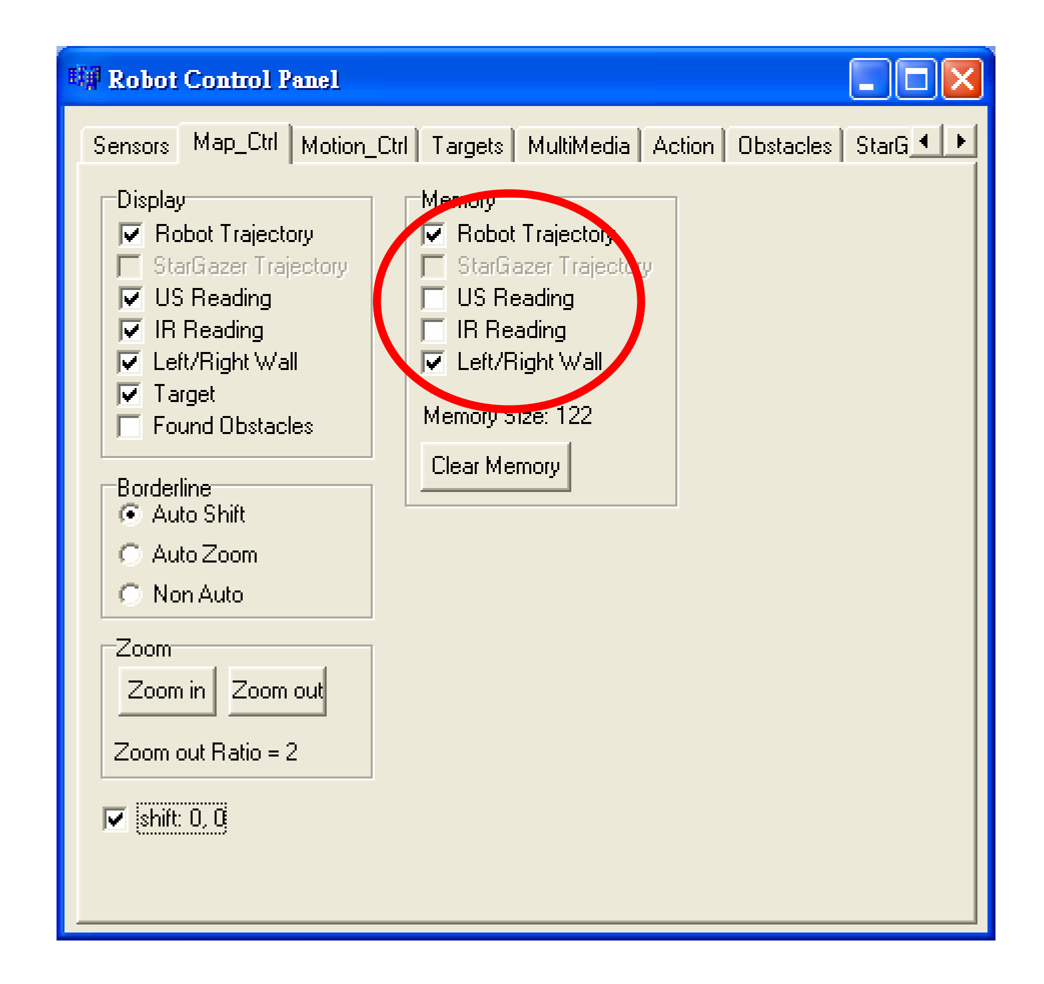


- Save Obstacles Position

儲存地圖資訊，副檔名為.map。地圖資訊包括(1)機器人軌跡、(2)定位系統軌跡、(3)

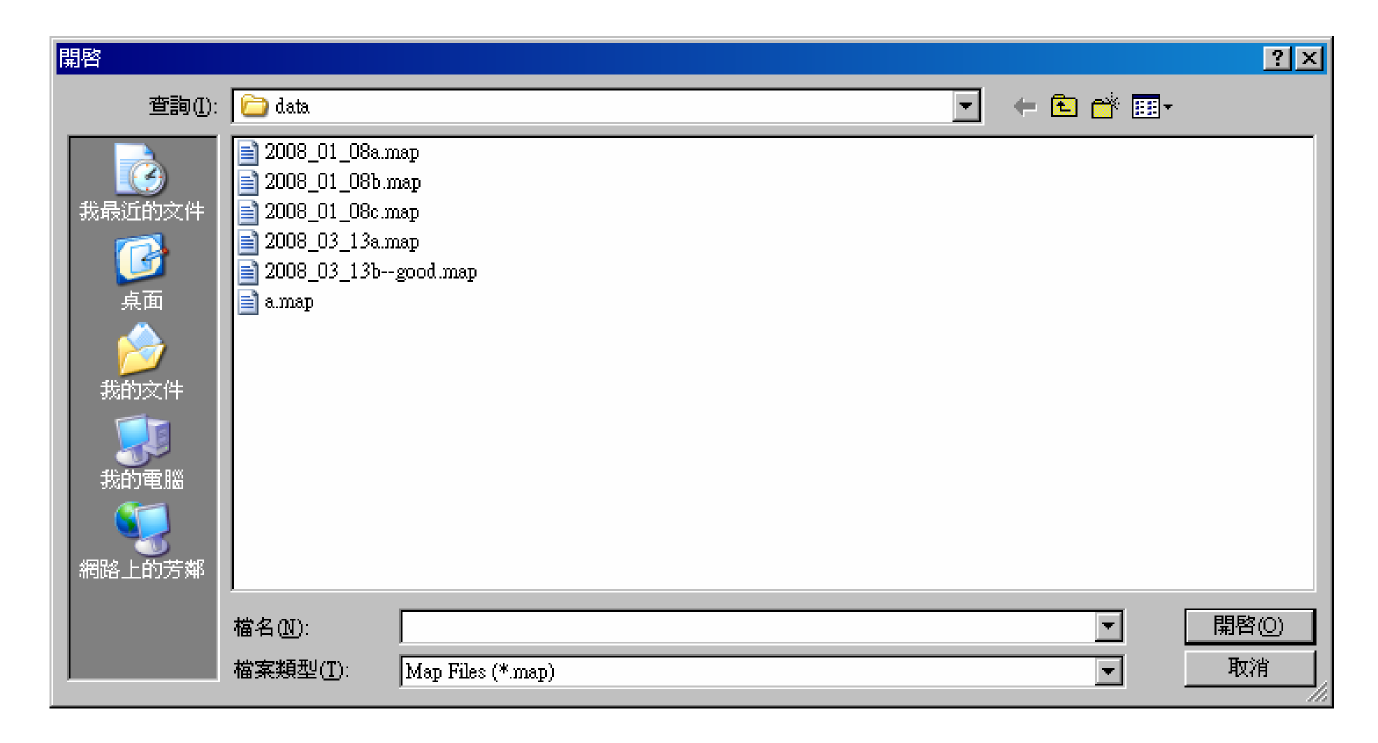
超音波讀到的障礙物、(4)紅外線讀到的障礙物、(5)左右牆的位置，如下圖：





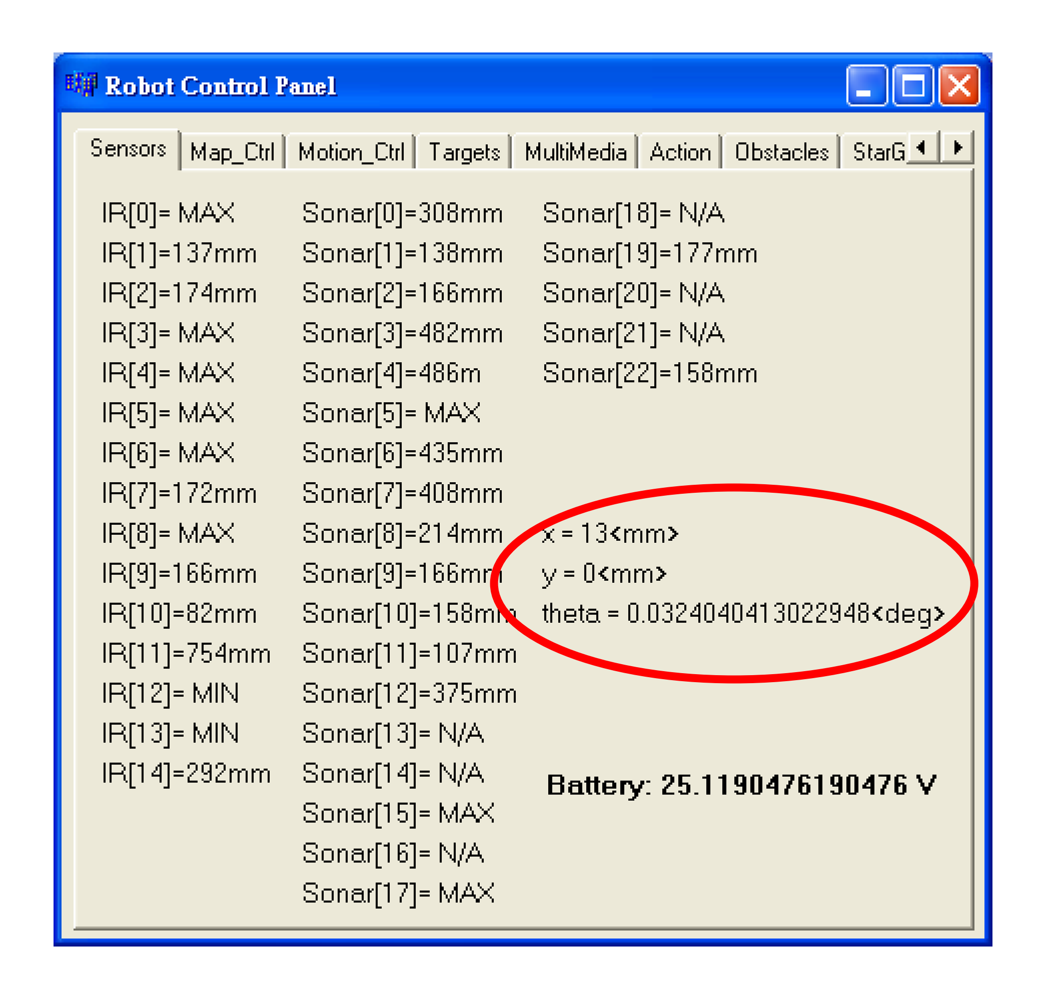
- Load Obstacles Position

載入地圖資訊，副檔名為.map。地圖資訊包括(1)機器人軌跡、(2)定位系統軌跡、(3)超音波讀到的障礙物、(4)紅外線讀到的障礙物、(5)左右牆的位置，如下圖：



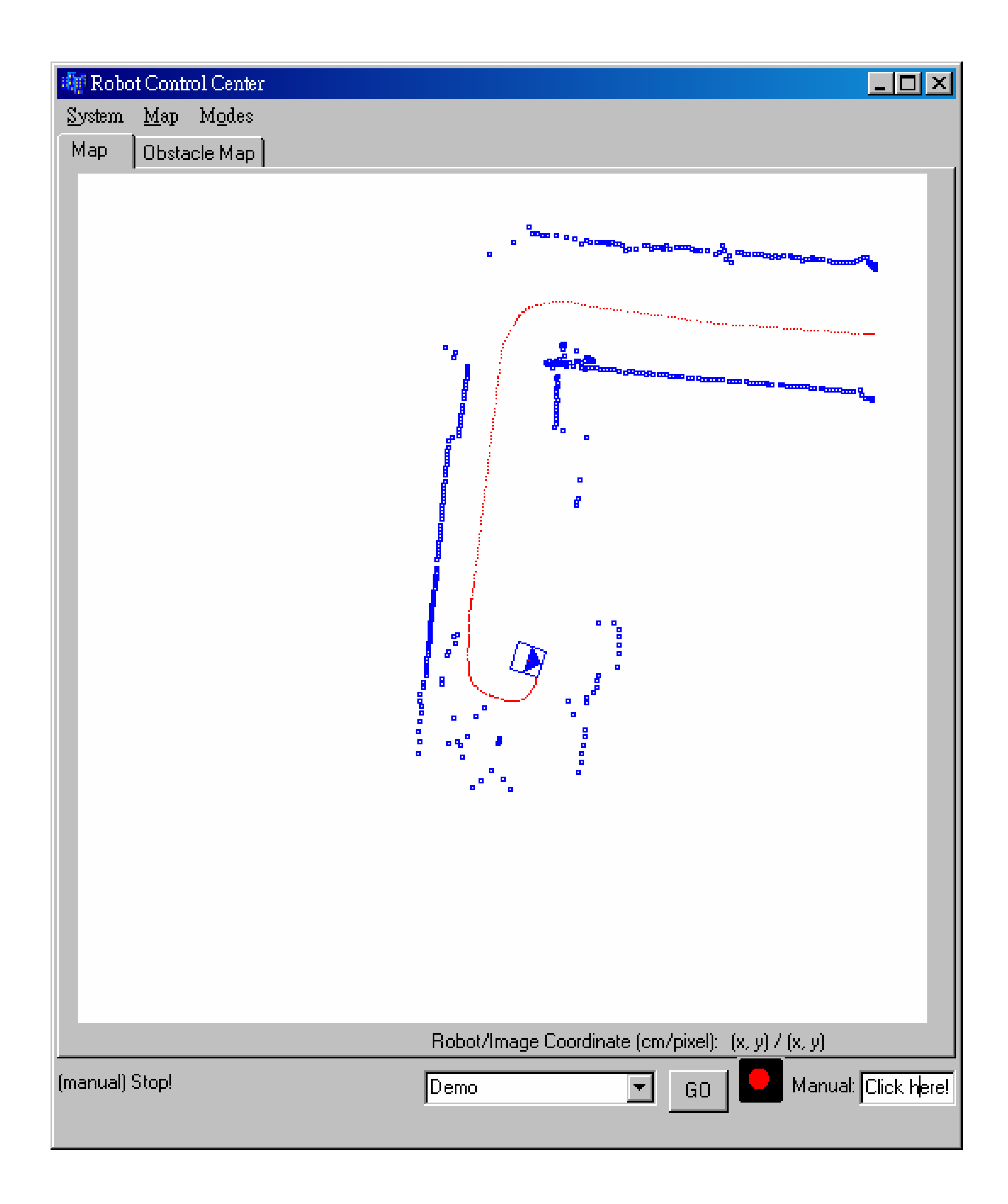
- Reset to Origin

將機器人目前座標設定到座標原點(0, 0)，如下圖：

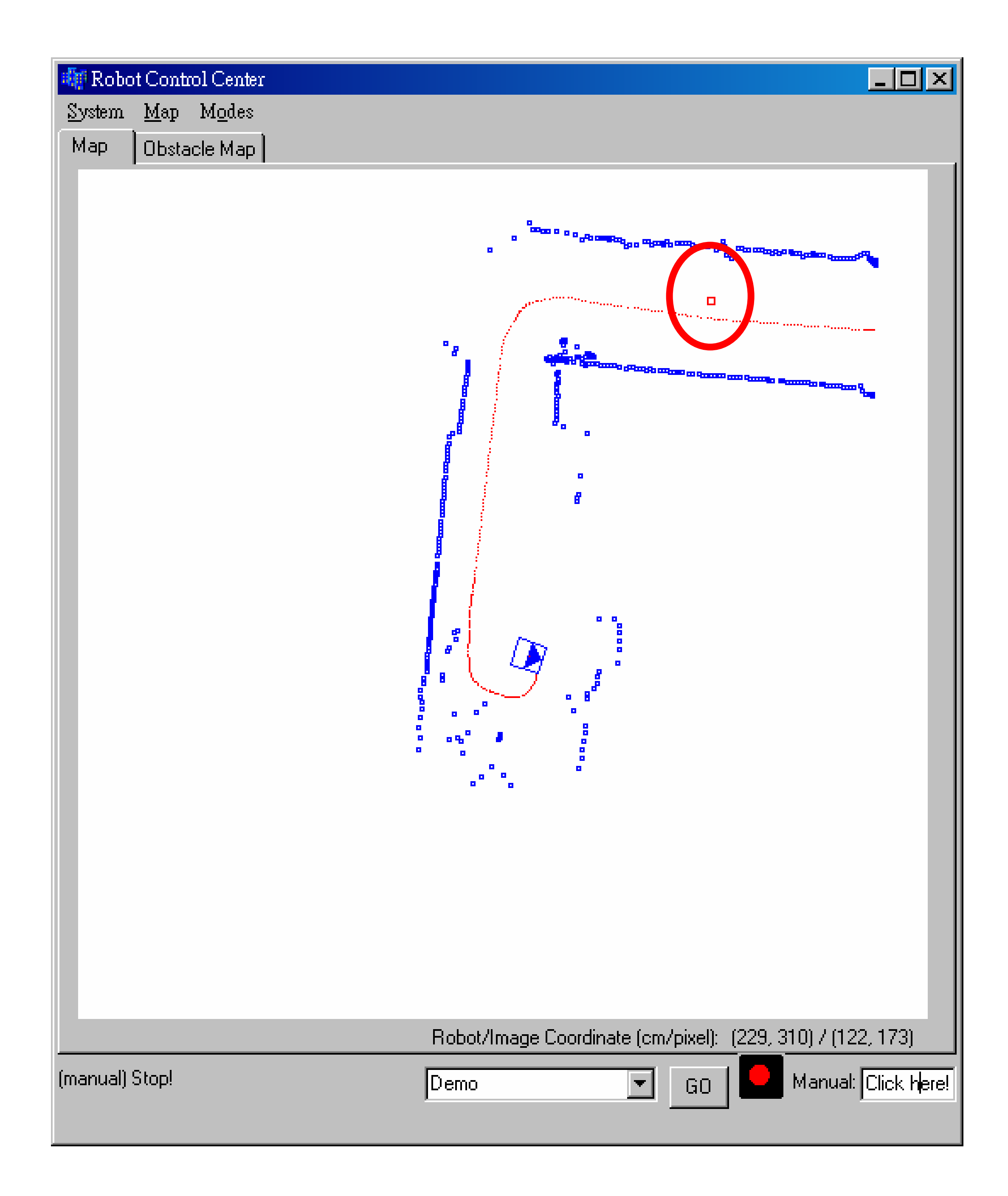


- Path Planning

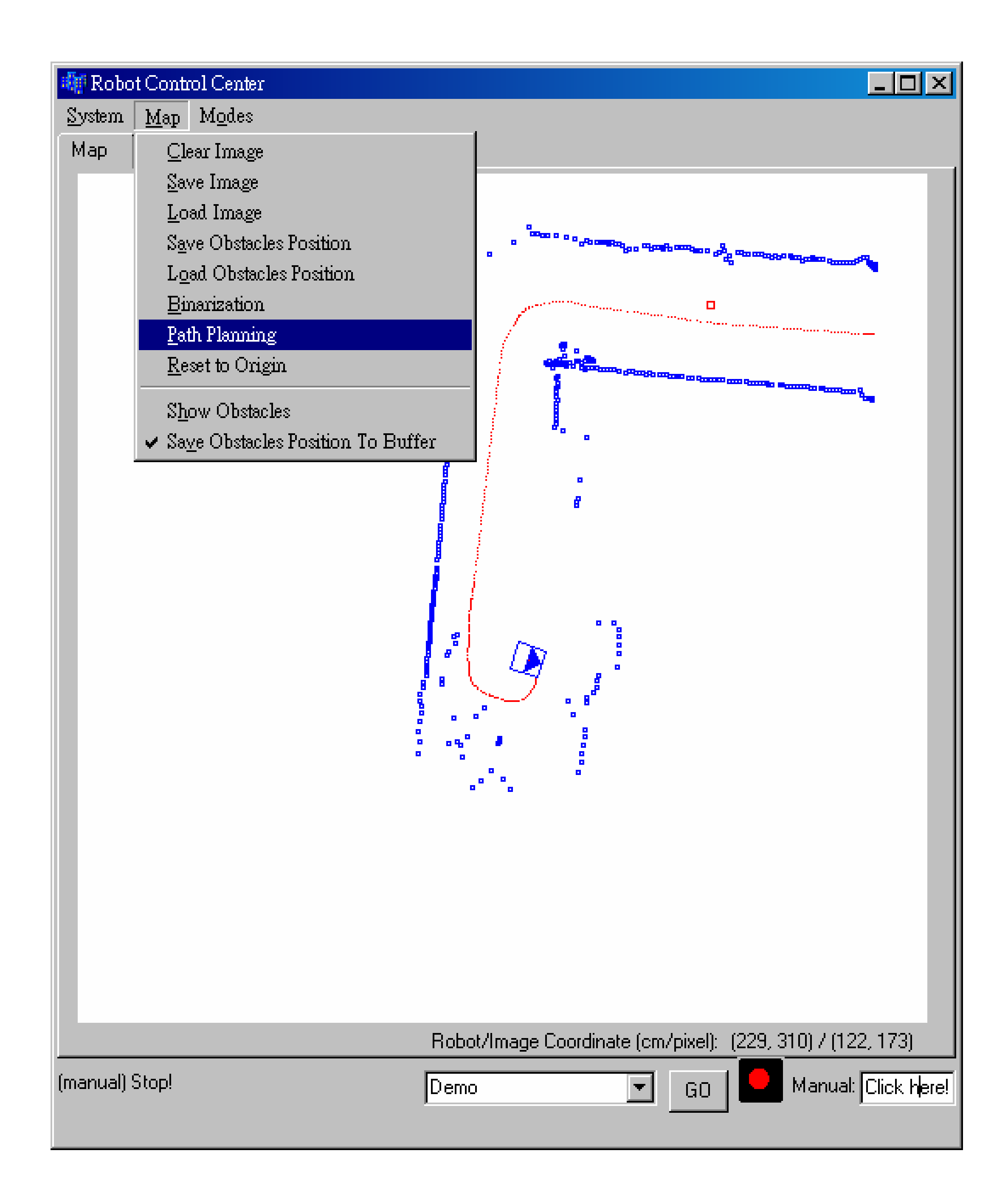
  指定目的地之後，此功能將自動偵測機器人目前位置、目的地位置、地圖環境，以決定出應走的路徑，並成功抵達目的地。如下圖：


Robot Control Center
System
Map
Modes
Map
Obstacle Map
Robot/Image Coordinate (cm/pixel): (x, y) / (x, y)
(manual) Stop!
Demo
GO
Manual:
Click Here!

滑鼠點擊目的地：

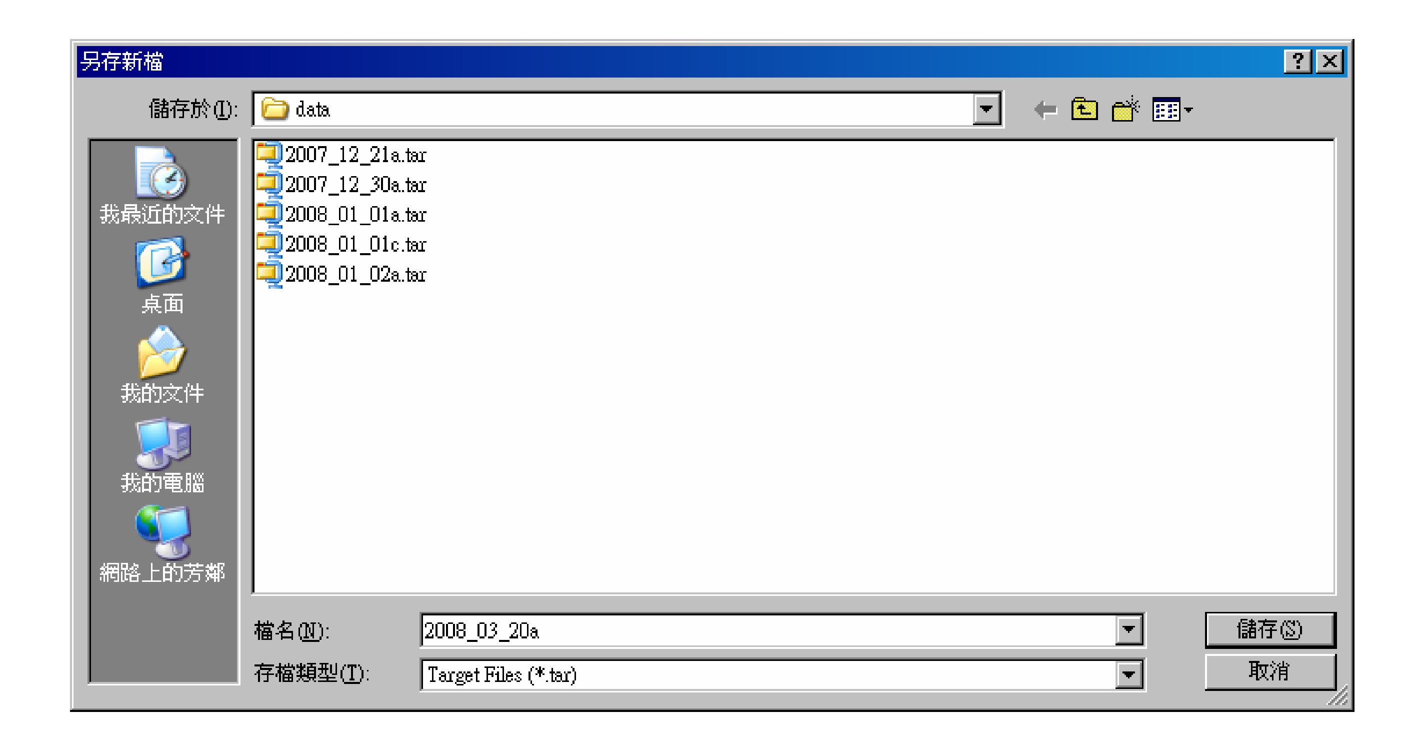


選 Path Planning：

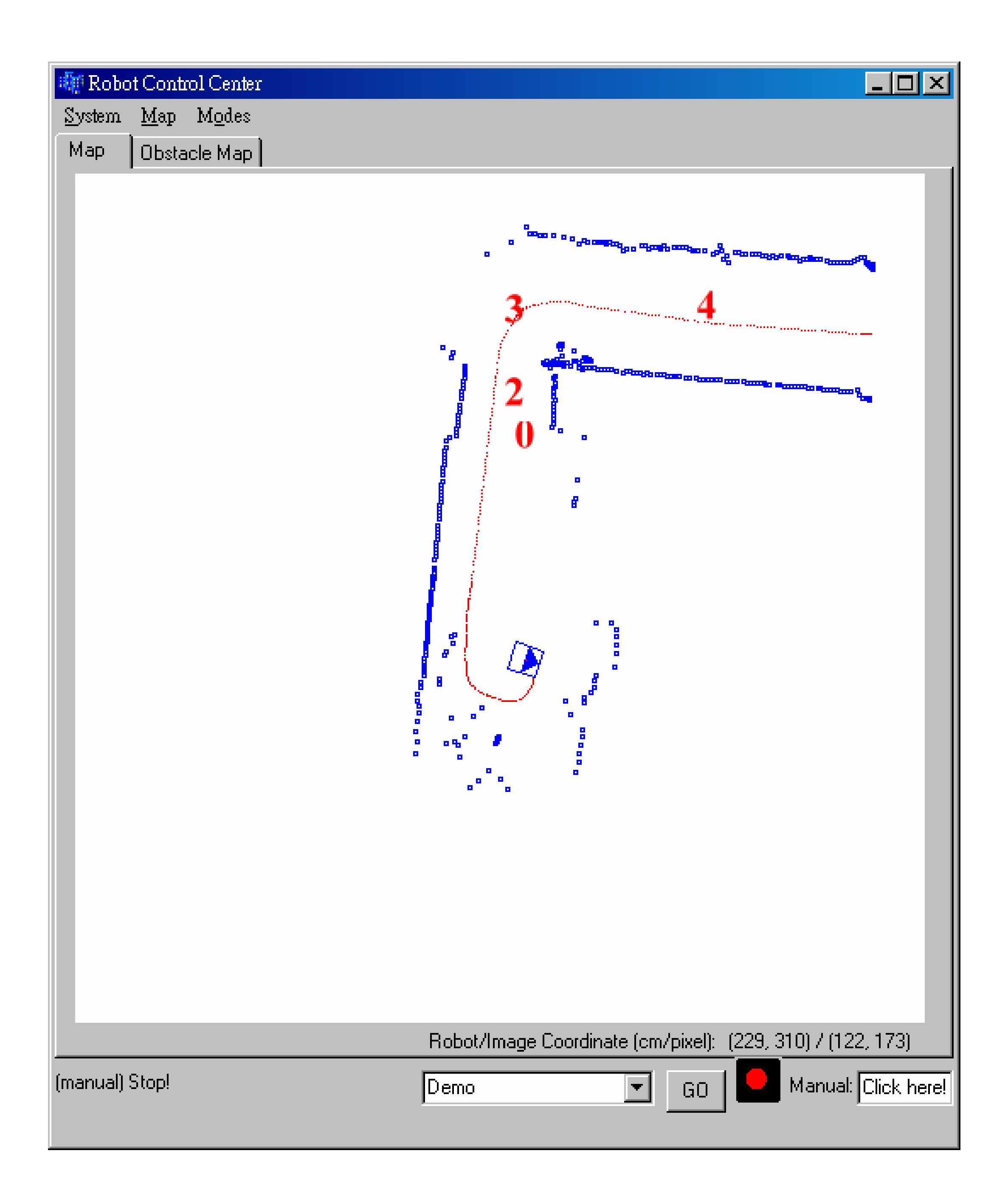


Robot Control Center
System
Map
Modes
Map
Clear Image
Save Image
Load Image
Save Obstacles Position
Load Obstacles Position
Binarization
Path Planning
Reset to Origin
Show Obstacles
Save Obstacles Position To Buffer
Robot/Image Coordinate (cm/pixel): (229, 310) / (122, 173)
(manual) Stop!
Demo
GO
Manual:
Click Here!

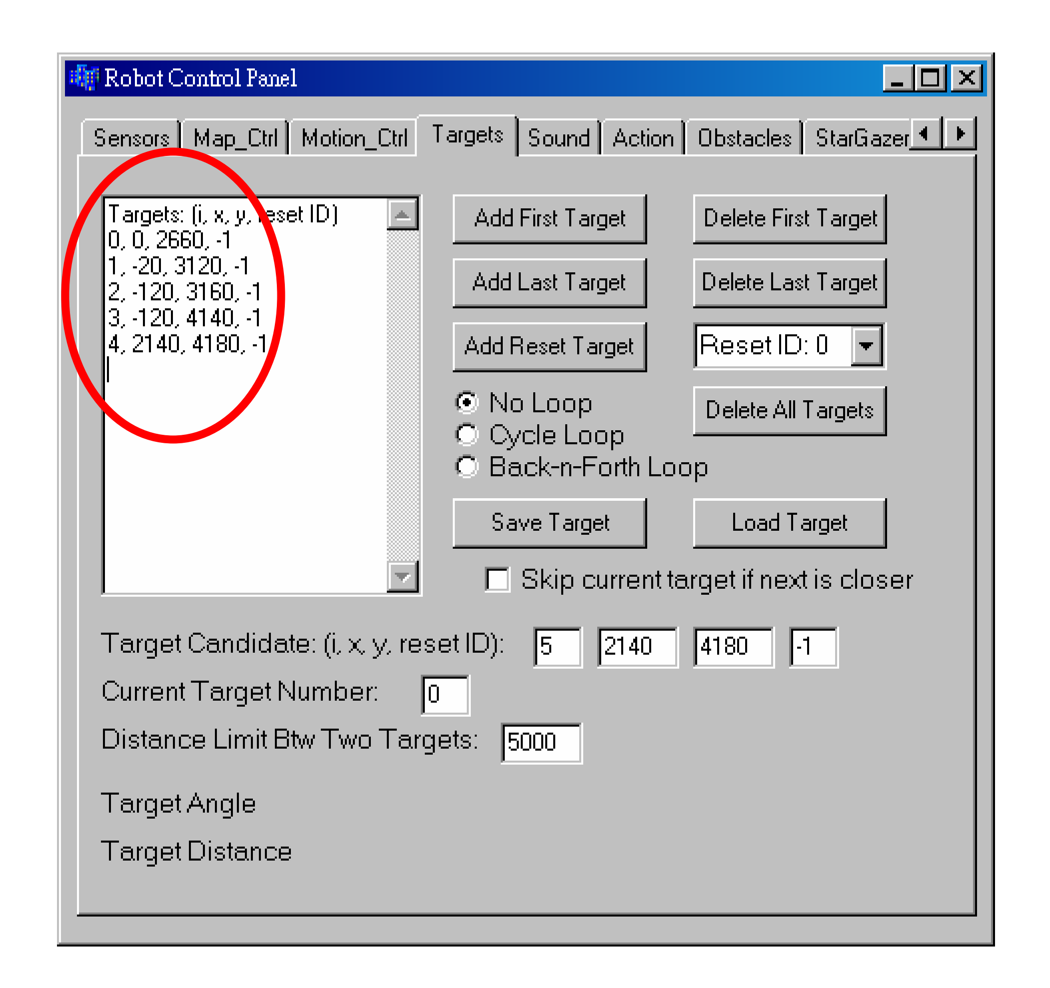
將路徑規劃結果存檔：



路徑規劃已決定出路徑，並標示於地圖中：



並同時將路徑轉折點資訊列表：

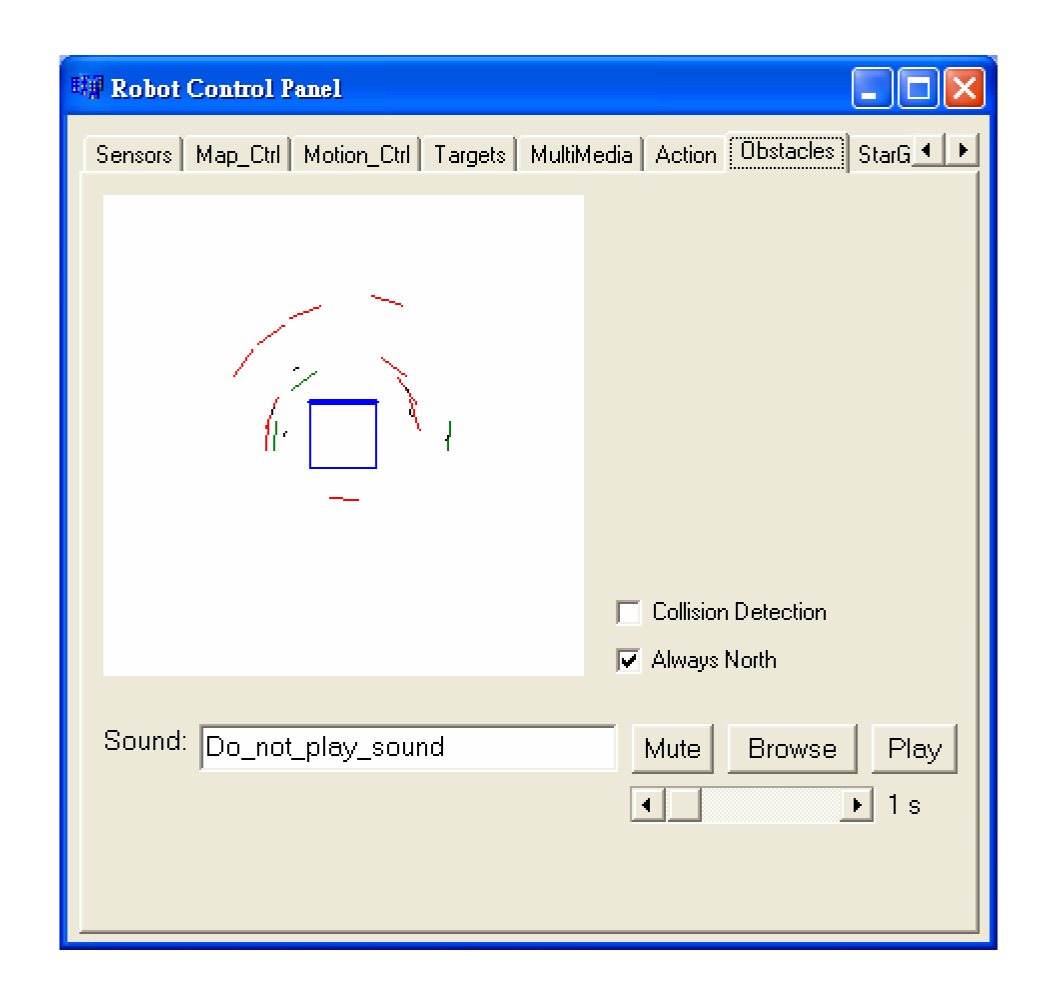


此路徑規劃乃是針對目前地圖區塊的環境設計。若是發現地圖區塊中無法找到可行之路徑以移動到目的地，則會顯示路徑規劃失敗，如下圖：



- Show Obstacles

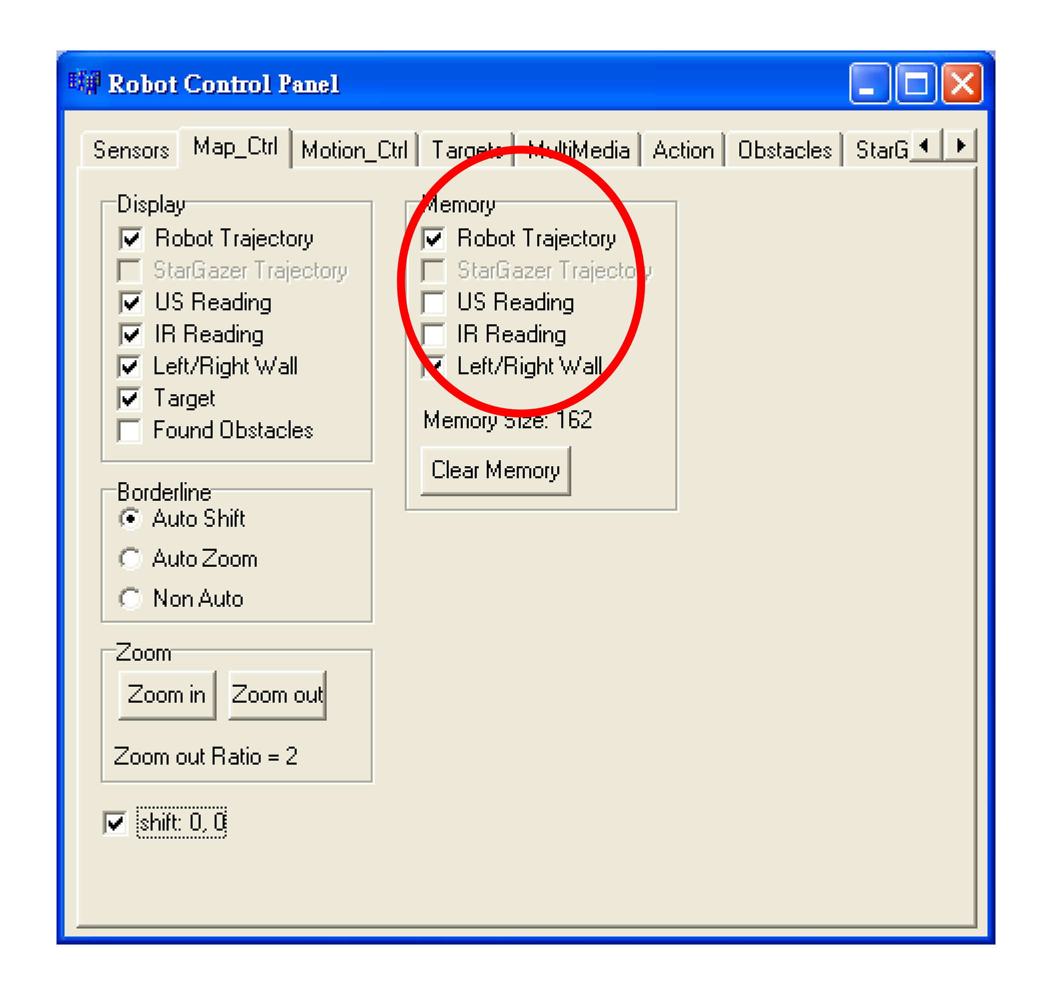
  將目前感測器偵測到的障礙物相對位置繪製出來，繪在資訊區塊的 Obstacles 頁面。如下圖：



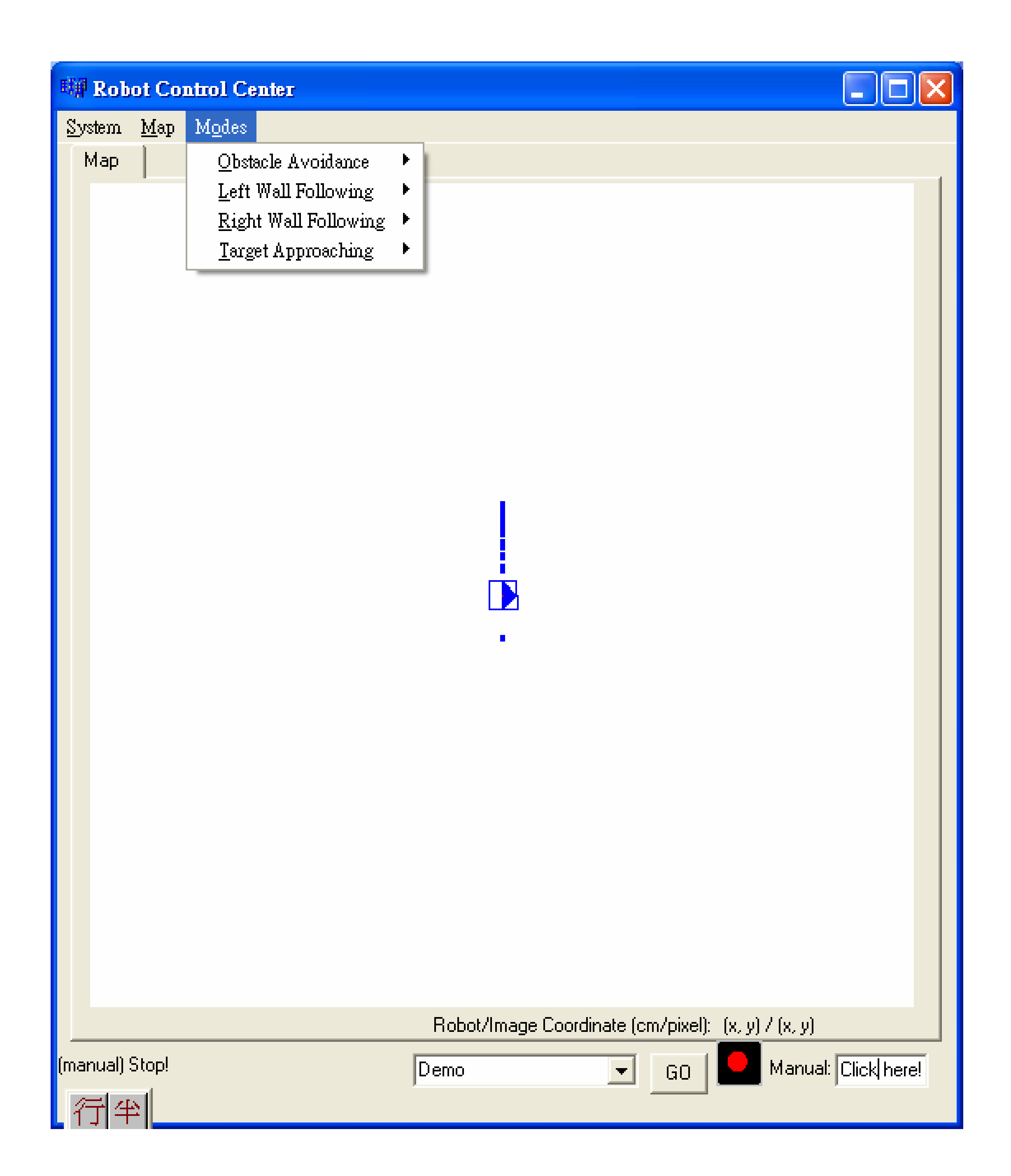
其中，中間藍色正方形為機器人本體，粗藍線代表機器人的前方。黑色短線代表紅外線偵測到的障礙物，紅色長線代表底盤超音波偵測到的障礙物，綠色長線代表上盤超音波偵測到的障礙物。另外，當所偵測到的障礙物為地圖中已有的障礙物時(也就是靜態障礙物)，該線段會以粗體表示(如圖片中下方的短粗黑線就代表紅外線所偵測到的靜態障礙物)；反之，若是為動態障礙物，則線段會以細線表示。若要停止顯示障礙物位置，則將打勾解除即可(再按一次此功能)。要注意須先執行 Map→Binarization 才能正確偵測到靜態障礙物。

- Save Obstacles Position to Buffer

  將地圖資訊儲存至記憶體中。若要停止儲存地圖資訊至記憶體中，則將打勾解除即可(再按一次此功能)。


Robot Control Panel
Sensors
Map_Ctrl
Motion_Ctrl
Targets
MultiMedia
Action
Obstacles
StarG
Display
Robot Trajectory
StarGazer Trajectory
US Reading
IR Reading
Left/Right Wall
Target
Found Obstacles
Borderline
Auto Shift
Auto Zoom
Non Auto
Zoom
Zoom in
Zoom out
Zoom out Ratio = 2
shift: 0, 0
Memory
Robot Trajectory
US Reading
IR Reading
Left/Right Wall
Memory Size: 162
Clear Memory

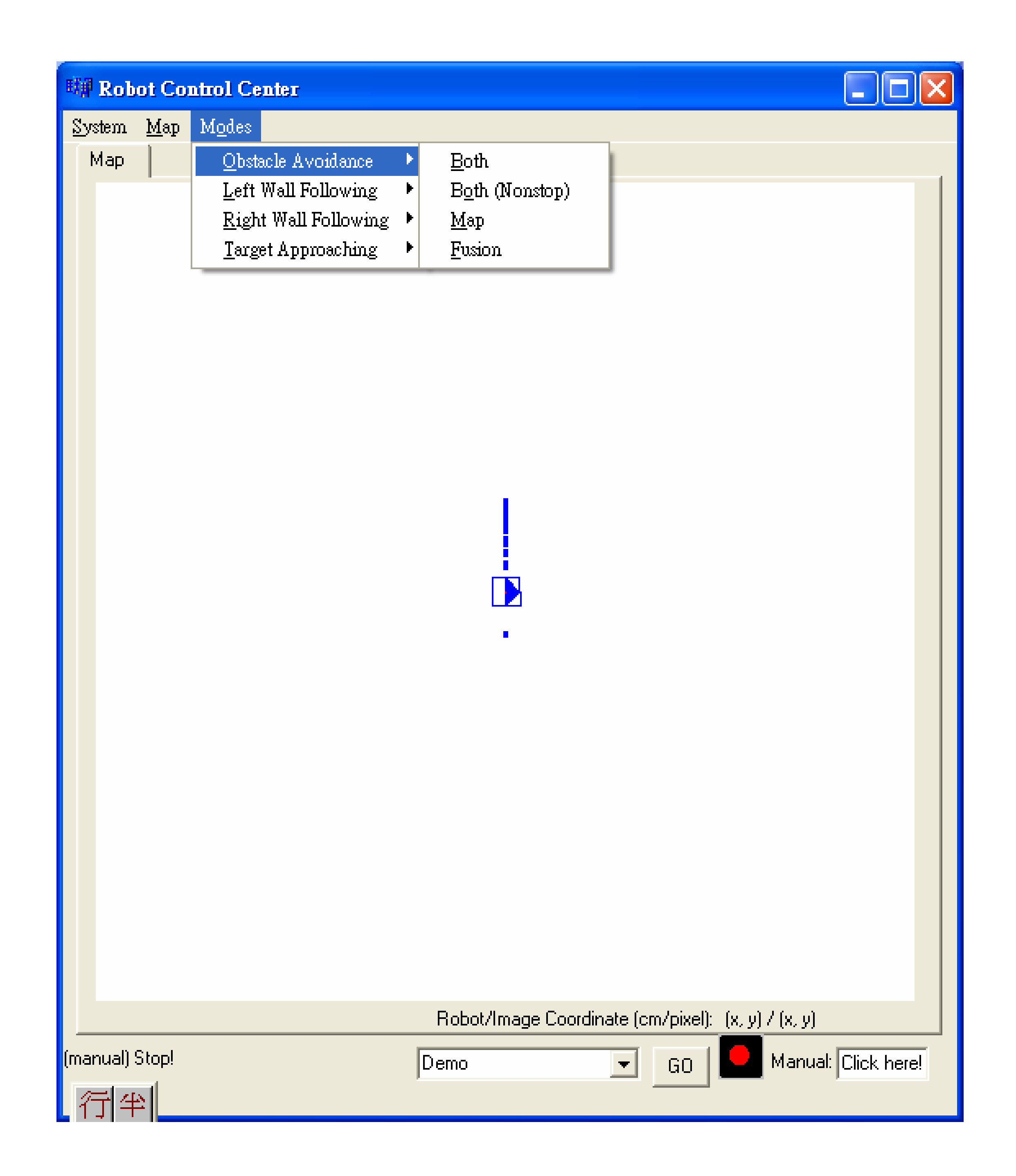
3. Modes



Modes 選單中有以下功能：

- Obstacle Avoidance

此功能純粹作避障用，尤其是使用鍵盤、搖桿、或是遠端操控機器人時搭配使用。一旦偵測到障礙物，機器人會發出喇叭聲警示。如下圖：



- Both

執行此功能後，當感測器偵測到前方有障礙物時，會以原地旋轉的方式來避障。若是障礙物在機器人右前方，則機器人會原地左轉來避障；反之，若障礙物在機器人左前方，則機器人會原地右轉來避障。若是後方感測器偵測到有障礙物，且使用鍵盤或搖桿遙控機器人往後退時，機器人會停止(不執行往後退命令)。此功能會一直持續進行避障，直到打勾解除為止(再按一次此功能)。

- Both (Nonstop)

此功能與 Both 非常類似，唯一的差別在於看到障礙物之後，是採取停止的方式，而不是如 Both 是以原地旋轉的方式來避開障礙物。此功能最主要的使用時機在於導覽用途，且此功能會一直持續進行避障，直到打勾解除為止(再按一次此功能)。

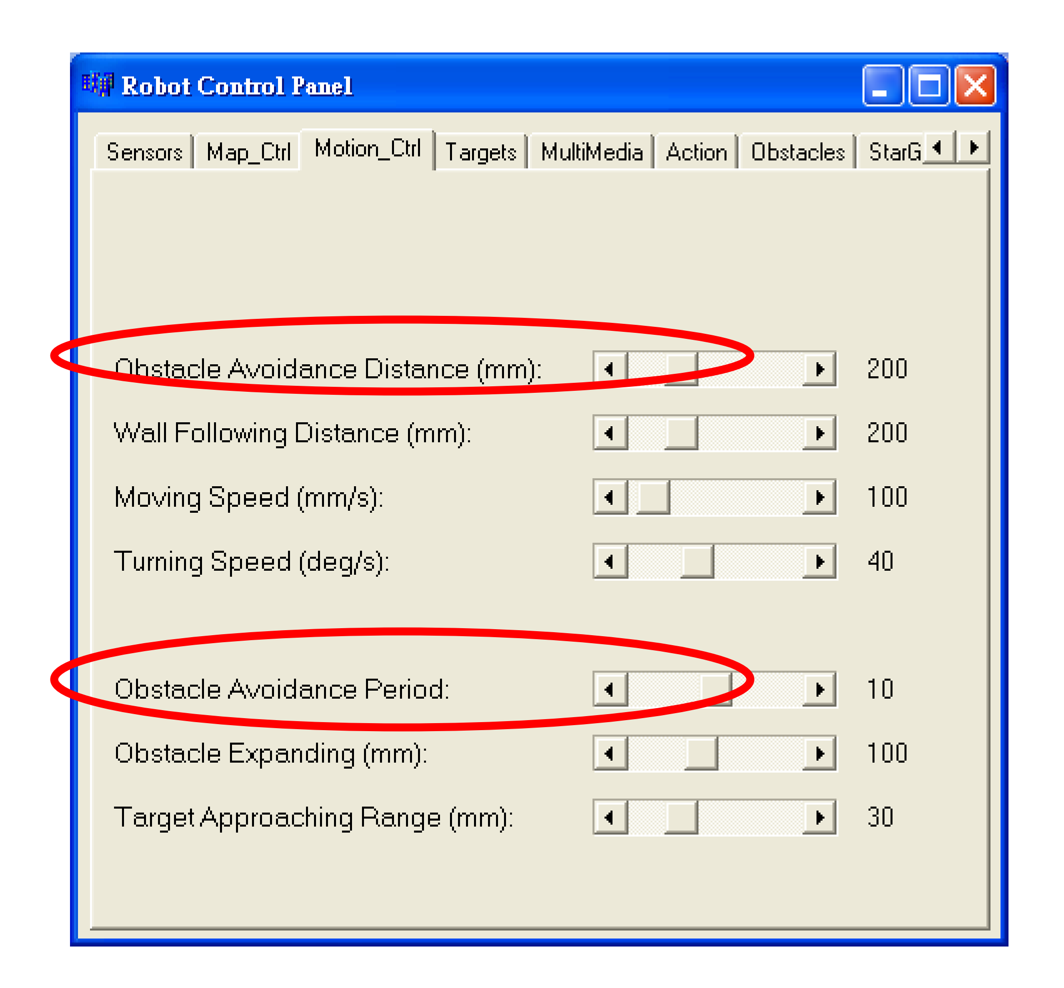
- Map

執行此功能後，當偵測到地圖區塊中機器人前方有障礙物時，會以原地旋轉的方式來避障。若是障礙物在機器人右前方，則機器人會原地左轉來避障；反之，若障礙物在機器人左前方，則機器人會原地右轉來避障。若是後方偵測到地圖區塊中有障礙物，且使用鍵盤或搖桿遙控機器人往後退時，機器人會停止。此功能會一直持續進行避障，直到打勾解除為止(再按一次此功能)。請注意此功能建議搭配定位系統使用，否則可能因為機器人本身的座標系統和實際空間座標系統的誤差過大，導致避障效果降低。

- Fusion

此功能為以上兩種功能的綜合，也就是無論是感測器或是在地圖區塊中偵測到機器人前方有障礙物時，皆會以原地旋轉的方式來避障。若是障礙物在機器人右前方，則機器人會原地左轉來避障；反之，若障礙物在機器人左前方，則機器人會原地右轉來避障。若是後方偵測到有障礙物，且使用鍵盤或搖桿遙控機器人往後退時，機器人會停止(不執行往後退命令)。此功能會一直持續進行避障，直到打勾解除為止(再按一次此功能)。請注意此功能建議搭配定位系統使用，否則可能因為機器人本身的座標系統和實際空間座標系統的誤差過大，導致避障效果降低。

避障功能有兩項參數可以設定，如下圖：



1. Obstacle Avoidance Distance (避障距離，單位：mm)

   也就是設定機器人偵測到障礙物在前方多少距離時，必須執行避障動作。

2. Obstacle Avoidance Period

   此參數是為了解決遇到”八”字形區域時會用到的參數。因為若機器人看到”八”字形的左邊時，會想要原地右轉避障；但是當機器人再往右繼續旋轉，而看到”八”字形的右邊時，會發現右邊有障礙物，因此會改成原地左轉避障，故而進入無窮迴圈。此值設定當機器人在發現障礙物進行原地旋轉後，一旦發現沒有障礙物時，要過多久的時間（緩衝期）才會再度啟用”障礙物在左邊就右轉避障，障礙物在右邊就左轉避障”的策略。若還在緩衝期就發現新障礙物，則無論此障礙物是在機器人左側或是右側，都會依之前的原地旋轉方向繼續避障，如此才能順利避開無窮迴圈的問題。若是旋轉速度設成越快，則此值應設成越高，反之亦然。

避障功能執行時可能會顯示的訊息如下：

- (oa) STOP!

  若使用鍵盤或是搖桿來遙控機器人，執行後退命令時，而後方超音波感測到有障礙物時，會停止後退。

- (oa) Left Turn!

  若使用鍵盤或是搖桿來遙控機器人，感測器發現機器人的右前方有障礙物時，會以原地左轉來避障。

- (oa_map) Left Turn!

  若使用鍵盤或是搖桿來遙控機器人，發現地圖區塊中機器人的右前方有障礙物時，會以原地左轉來避障。

- (oa) Right Turn!

  若使用鍵盤或是搖桿來遙控機器人，感測器發現機器人的左前方有障礙物時，會以原地右轉來避障。

- (oa_map) Right Turn!

  若使用鍵盤或是搖桿來遙控機器人，發現地圖區塊中機器人的左前方有障礙物時，會以原地右轉來避障。

- (oa) No Obstacles!

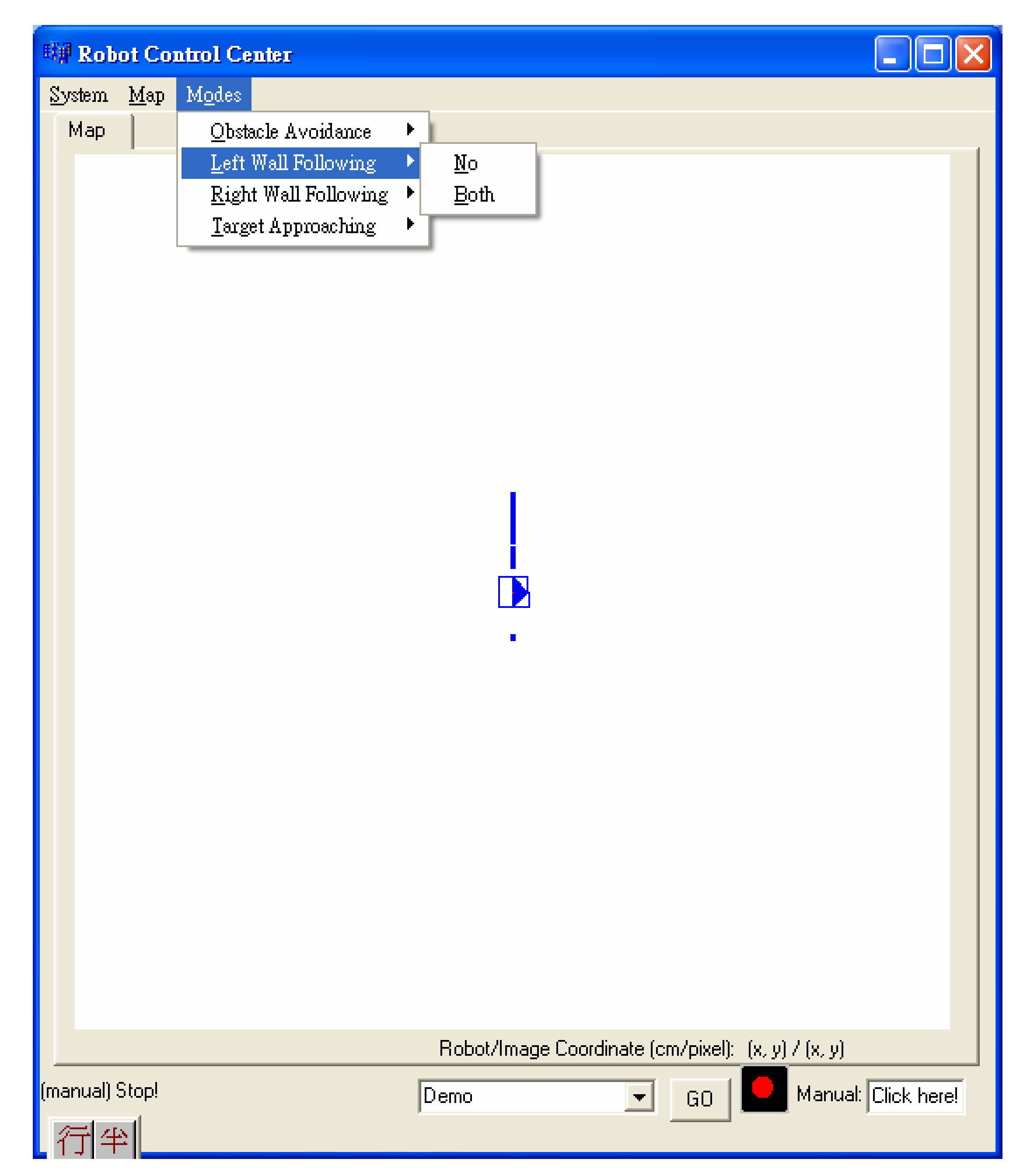
  若使用鍵盤或是搖桿來遙控機器人，沒有發現任何障礙物時，會停在原地不動等待遙控命令。

- (oa) Stop due to side oa!

  若使用鍵盤或是搖桿來遙控機器人，執行往左前方走或是往右前方走時，若發現機器人的左前側邊或右前側邊有障礙物離機器人很近時，會停止繼續往左前方走或是往右前方走。

- Left Wall Following

  此功能為沿著左牆移動。如下圖：



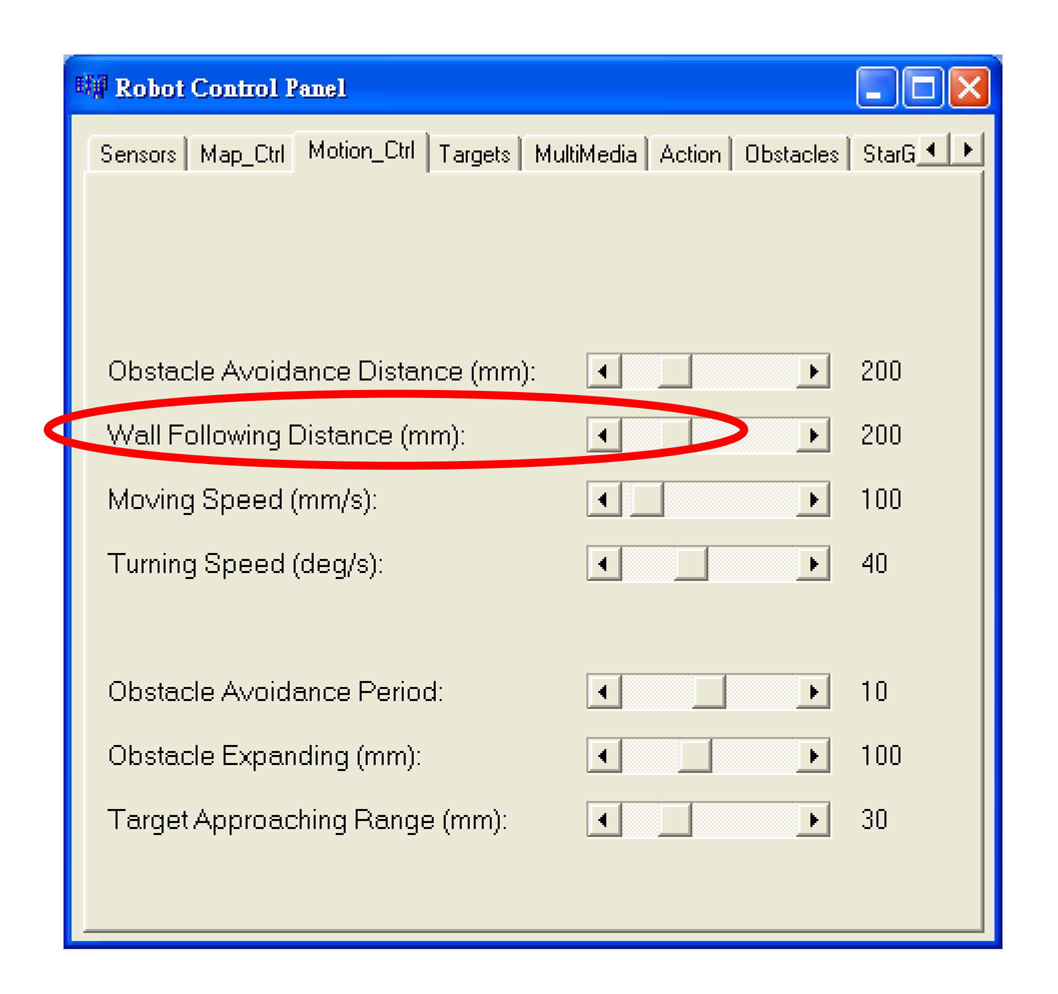
■ No

不避障之沿左牆走功能。若是要取消此功能，則將打勾解除即可(再按一次此功能)。

■ Both

當移動過程中發現有障礙物時，會以原地右轉避障之。若是前方已無障礙物之後，會跳回沿牆走的模式繼續下去，直到打勾解除為止(再按一次此功能)。可以搭配此功能進行自動地圖繪製。可以選擇用 No (不避障)或是用 Both(避障)。

可設定沿牆走時，要和牆壁保持的距離(Wall Following Distance)，單位為 mm。如下圖



沿左牆走功能執行時可能會顯示的訊息如下：

- (mode) Left Wall following (no)!

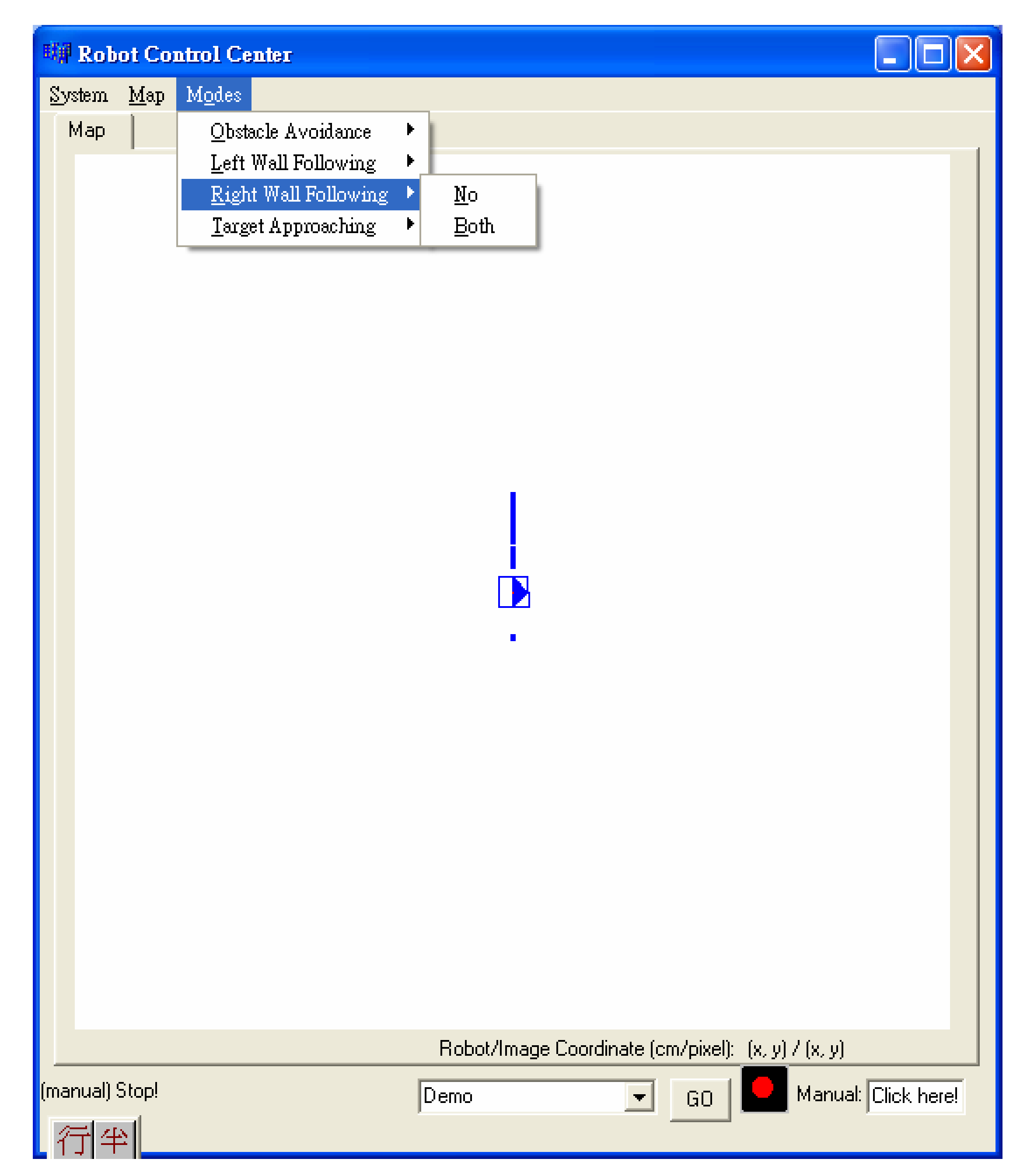
  不考慮避障時的沿左牆走模式。

- (mode) Left Wall following (oa)!

  考慮避障時的沿左牆走模式。

- Right Wall Following

  此功能為沿著右牆移動。此功能與 Left Wall Following 相仿，不同處是沿著右牆移動。如下圖：



■ No

不避障之沿右牆走功能。若是要取消此功能，則將打勾解除即可(再按一次此功能)。

■ Both

當移動過程中發現有障礙物時，會以原地左轉避障之。若是前方已無障礙物之後，會跳回沿牆走的模式繼續下去，直到打勾解除為止(再按一次此功能)。可以搭配此功能進行自動地圖繪製。和牆壁保持的距離參數(Wall Following Distance)與沿左牆走模式共用。

沿右牆走功能執行時可能會顯示的訊息如下：

- (mode) Right Wall following (no)!

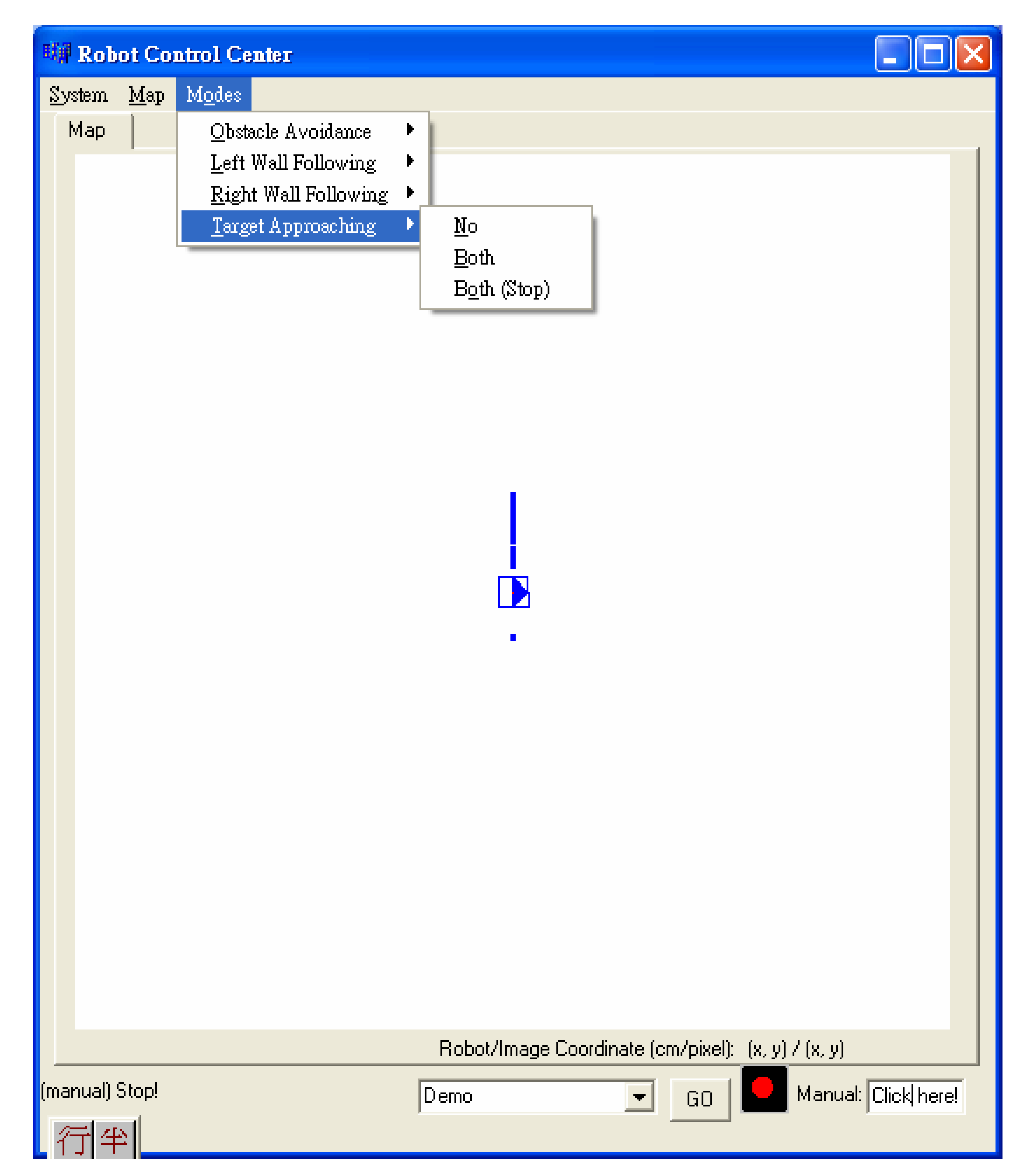
  不考慮避障時的沿右牆走模式。

- (mode) Right Wall following (oa)!

  考慮避障時的沿右牆走模式。

- Target Approaching

  此為目標點移動功能。可設定一個或多個目標點，機器人可以依照順序一個接著一個前往。預設的目標點移動模式是機器人會往目前所給定的目標點移動，若是機器人和目標點的距離小於 Target Approaching Range 時，就當作機器人已成功抵達目前的目標點。若此為最終之目標點，則機器人會停在此目標點上，並發訊息”已抵達目標點，完成任務”。若是此目標點並非最終之目標點，則機器人會更新目前所前往之目標點資訊，並開始前往之。如下圖：



■ No

不避障之目標點移動功能。若是要取消此功能，則將打勾解除即可(再按一次此功能)。

■ Both

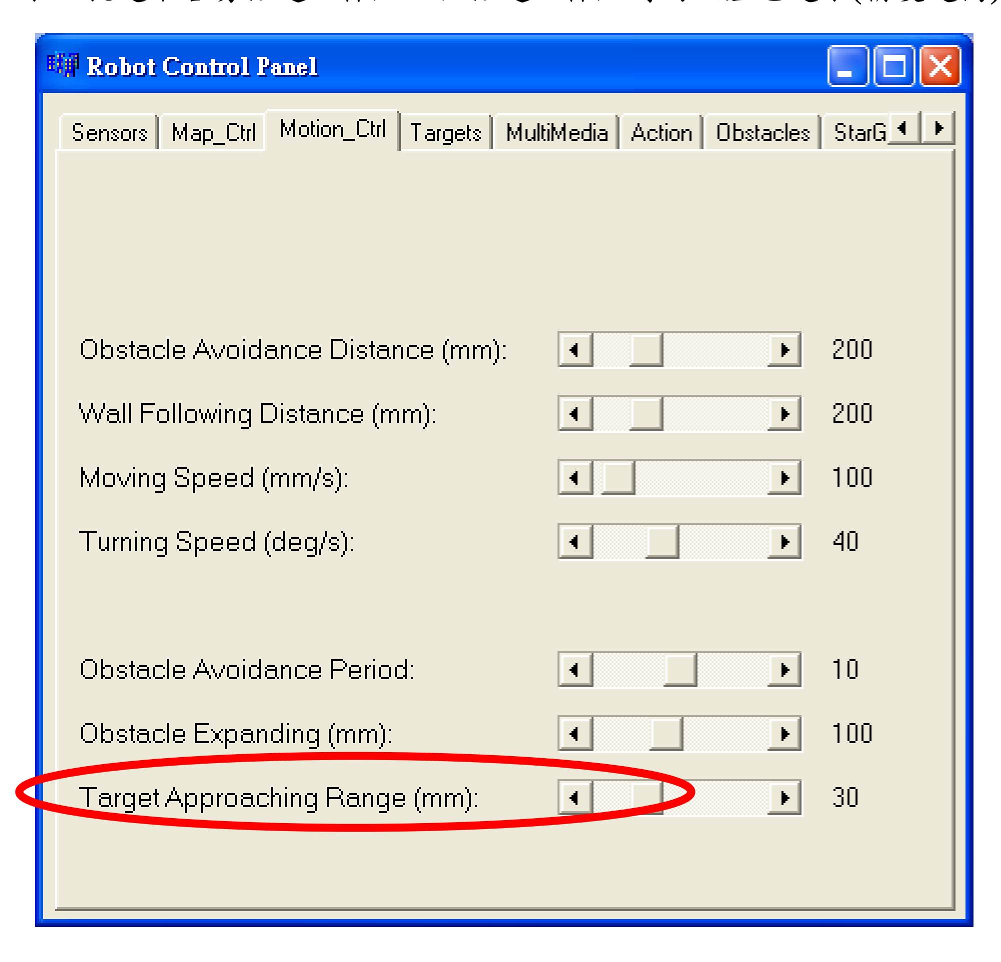
需避障之目標點移動功能。若是要取消此功能，則將打勾解除即可(再按一次此功能)。

■ Both (Stop)

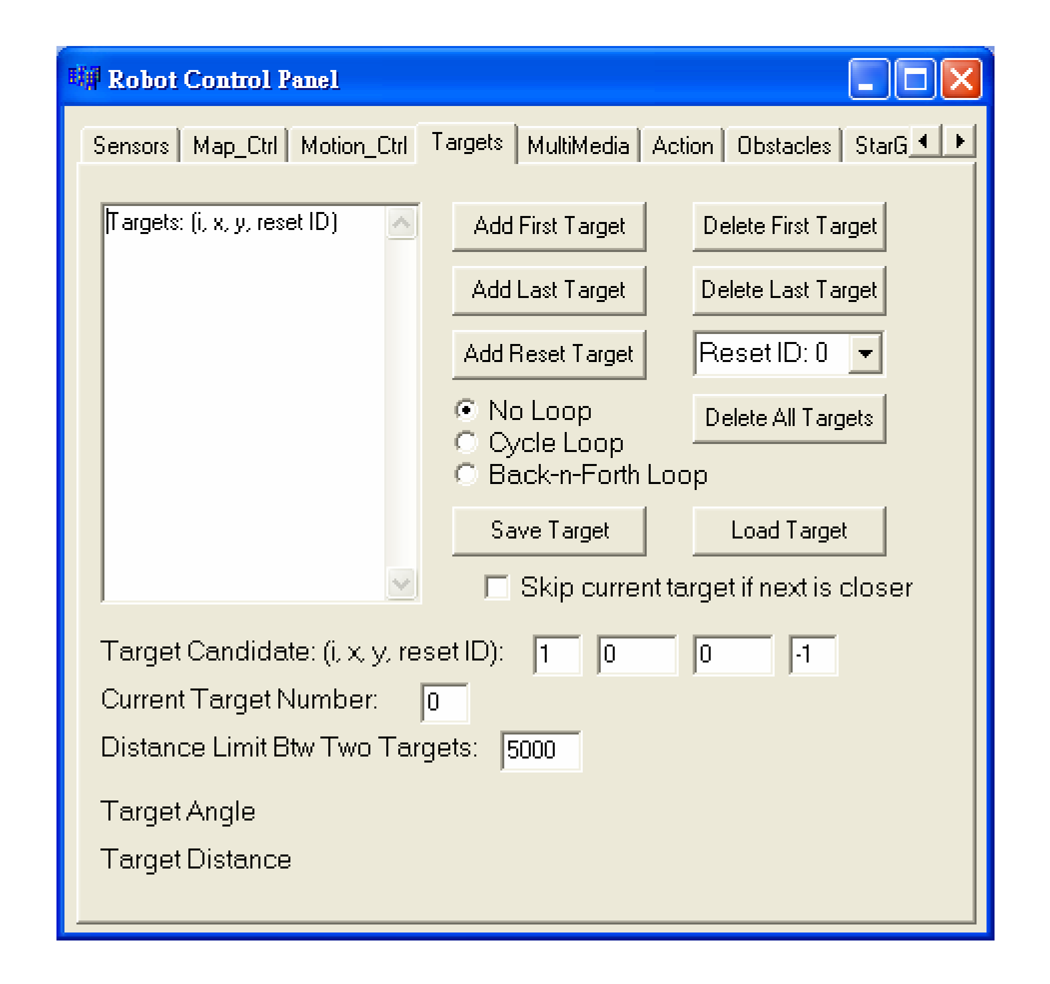
同樣是需避障之目標點移動功能，差異在於偵測到障礙物時會停止，而不是繞

開它。此功能適合用於機器人移動路徑中僅有會移動的物體(如人)時。若是要取消此功能，則將打勾解除即可(再按一次此功能)。

可設定所謂目標點抵達與否的距離(Target Approaching Range)，單位為 mm。此距離設得越大，能夠越容易抵達目標點，但抵達目標點時的誤差也越大(精度越低)。反之，若此距離設得越小，就越不容易抵達目標點，但抵達目標點時的誤差也越小(精度越高)。如下圖：



在目標點移動功能中，可以選擇和調整的選項/參數的功能較多，如下圖：



- 目標點列表

  將已儲存在記憶體中的目標點列表顯示出來。會顯示的目標點資訊包含：編號、x 座標、y 座標、以及 Reset ID (若是該目標點並非 Reset 點，則 ID 顯示-1)。

- Add First Target

  增加一新目標點(此新目標點插至最前面)。

- Add Last Target

  增加一新目標點(此新目標點放至最後面)。

- Add Reset Target

  增加一新 Reset 點(此新 Reset 點放至最後面)。Reset 點目前是使用 Evolution Robotics 的產品：NorthStar 定位系統來輔助機器人做座標系統的校正。由於機器人使用 Dead Reckoning，也就是利用馬達回授編碼器資訊，來做座標系統的定義。因此，不可避免地，會有累積誤差的問題。尤其是在行走的地面不平坦或是在地面材質較光滑時。此時，使用目標點移動模式，長時間可能導致機器人本身的座標系統和實際空間座標系統的誤差

過大，而無法到達一開始設定的目標點。但是若在所要到達的目標點中，有使用 NorthStar 定位系統來當作 Reset 點(同時也是其中一個目標點)，就能夠大幅減少此問題的發生機會。

- Reset ID 設定

  在 Add Reset Target 之前須先設定要新增的 Reset 點之 ID。ID 數值是在 NorthStar 定位系統板上設定。

- Delete First Target

  刪除一舊目標點(在目標點列表中最前面的目標點)。

- Delete Last Target

  刪除一舊目標點(在目標點列表中最後面的目標點)。

- Delete All Targets

  刪除全部的目標點。

- 目標點移動選項：

  - No Loop

    也就是不會無止盡地依循目標點移動。一旦機器人走完所設定的所有目標點後，會停止在最後一個目標點處。假如有 5 個目標點，則機器人的移動順序是：

    0→1→2→3→4

  - Cycle Loop

    會依所設的目標點以循環的方式持續移動。當機器人走到所設定的最後一個目標點後，會持續朝第一個目標點前進。假如有 5 個目標點，則機器人的移動順序是：

    0→1→2→3→4→0→1→2→3→4→0→…

  - Back-n-Forth Loop

    當機器人走到所設定的最後一個目標點後，會折回來繼續走(朝倒數第二個目標點前進，順序變成反過來)，以此類推。假如有 5 個目標點，則機器人的移動順序是：

    0→1→2→3→4→3→2→1→0→1→…

- Skip current target if next is closer

  預設的目標點移動模式是機器人會往目前所給定的目標點移動，若是目前正在前往的目

標點被障礙物佔據時，機器人會偵測到障礙物並試圖避開(在 Both 或 Dynamic 模式)。但是若此障礙物一直不移開時，則機器人可能會不停地繞而無法抵達此目標點。但若此目標點並非最終之目標點時，則可以選取此功能。也就是說，機器人會試圖偵測在適當的時機，跳過目前的目標點，而直接向下一個目標點移動。

- Target Candidate

  目標點的選取方法，是使用滑鼠在地圖上直接點擊。請注意，若是當 Control Panel→Map_Ctrl→Display→Target 沒有勾選時，滑鼠點擊地圖是無法點選 Target Candidate 的。當滑鼠左鍵按到地圖的任何一個位置時，此處都會顯示出該位置的資訊，包含：

  (1) 目標點編號

  (2) 此目標點 x 軸數值 (單位：mm)

  (3) 此目標點 y 軸數值 (單位：mm)

  (4) 此目標點的 Reset ID

- Current Target Number

  此處顯示機器人正在前往的目標點編號。

- Distance Btw Two Targets

  此參數代表任意連續兩個目標點間的最大距離。若是在新增目標點時，和上一個目標點的距離超過此數值時，會顯示警告視窗，提醒您重新設定目標點的位置(新目標點應設得離上一個目標點近一些)。此參數也代表著，若機器人在目標點移動過程中，有發現障礙物並試圖避開時，會依據此參數來偵測是否因避障而繞得太遠。

- Target Angle

  此處顯示機器人目前的朝向角和目標點的角度差。

- Target Distance

  此處顯示機器人目前的位置和目前正在前往的目標點間的距離。

目標點移動功能執行時可能會顯示的訊息如下：

- (mode) Nth Target Approaching (no)!

  不考慮避障時的目標點移動模式，同時顯示現在正在前往的目標點編號。

- (mode) Nth Target Approaching (both)!

  使用原地旋轉方式避障時的目標點移動模式，同時顯示現在正在前往的目標點編號。

- (mode) Nth Target Approaching (both_stop)!

  使用停止方式避障時的目標點移動模式，同時顯示現在正在前往的目標點編號。

- (mode) Target Arrived!

  已抵達最終目標點。任務完成！

- (mode) Left Wall following (ta)!

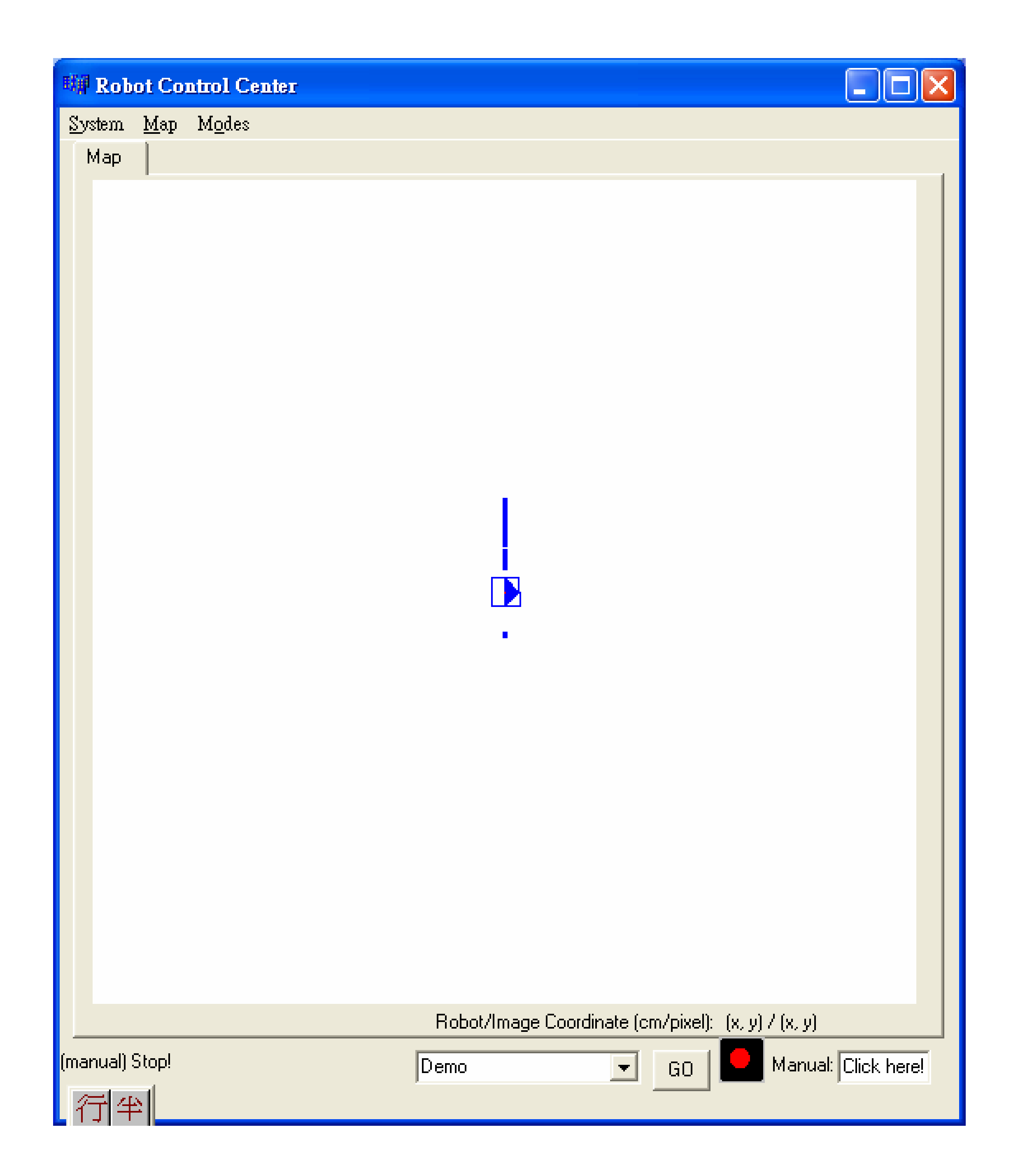
  選取 Both 模式來執行目標點移動功能時，當偵測到障礙物時，機器人會試圖閃避此障礙物並且沿著障礙物移動，同時也試圖能夠繼續目標點移動任務。此訊息表示機器人偵測到障礙物在其左側，因此使用沿左牆移動功能。

- (mode) Right Wall following (ta)!

  同上，此訊息表示機器人偵測到障礙物在其右側，因此使用沿右牆移動功能。

4. 訊息、遙控、與展示區塊

本章節中介紹訊息、遙控、與展示區塊的說明，他們皆在主介面中的下方欄位，如下圖：



1. 訊息

訊息欄顯示 MR-W1 目前的運動狀態。除了已在選單的 Modes 中介紹的各種訊息之外，此處介紹使用鍵盤或搖桿遙控機器人時會出現的訊息：

- (manual) Left turn!

  以鍵盤遙控 MR-W1 左轉。

- (manual) Straight!

  以鍵盤遙控 MR-W1 直走。

- (manual) Right turn!

以鍵盤遙控 MR-W1 右轉。

- (manual) Reverse!

  以鍵盤遙控 MR-W1 往後退。

- (manual) Forward-Right Turn!

  以鍵盤遙控 MR-W1 往右前方移動。

- (manual) Forward-Left Turn!

  以鍵盤遙控 MR-W1 往左前方移動。

- (manual) Stop!

  以鍵盤遙控 MR-W1 停止。

- (joystick) Left Turn!

  以搖桿遙控 MR-W1 左轉。

- (joystick) Straight!

  以搖桿遙控 MR-W1 前進。

- (joystick) Right Turn!

  以搖桿遙控 MR-W1 右轉。

- (joystick) Reverse!

  以搖桿遙控 MR-W1 後退。

- (joystick) Forward-Right Turn!

  以搖桿遙控 MR-W1 往右前方移動。

- (joystick) Forward-Left Turn!

  以搖桿遙控 MR-W1 往左前方移動。

- (joystick) Reverse-Right Turn!

  以搖桿遙控 MR-W1 往左後方移動。

- (joystick) Reverse-Left Turn!

  以搖桿遙控 MR-W1 往右後方移動。

- (joystick) Stop!

以搖桿遙控 MR-W1 停止。

2. 鍵盤遙控

若要使用鍵盤遙控 MR-W1 動作，請將滑鼠移至 “Click here!” 處，即可以使用上下左右鍵，來控制 MR-W1 進行前進、後退、原地左右轉等動作，並且還可以同時搭配 Ctrl、Alt、Shift 等鍵，詳列如下：

- ↑

  往前進

- ↓

  往後退

- ←

  原地左轉

- →

  原地右轉

- Ctrl + (↑或↓或←或→)

  用兩倍的速度移動

- Alt + (↑或↓或←或→)

  用三倍的速度移動

- Shift + ←

  往左前方移動

- Shift + →

  往右前方移動

3. 搖桿遙控

若要使用搖桿遙控 MR-W1 動作，請將 USB 搖桿插上 MR-W1，即可以使用上下左右鍵，來控制 MR-W1 進行前進、後退、原地左右轉等動作。另外，搖桿按鈕已被設定為以下動作：

- Button 1:

緊急煞車，同時取消正在執行的任務(功能如同 System→Stop)。

- Button 2:

  a. 速度改為 200 mm/s。

  b. 避障距離及沿牆走距離改為 20 cm。

- Button 3:

  a. 速度改為 400 mm/s。

  b. 避障距離及沿牆走距離改為 35 cm。

- Button 4:

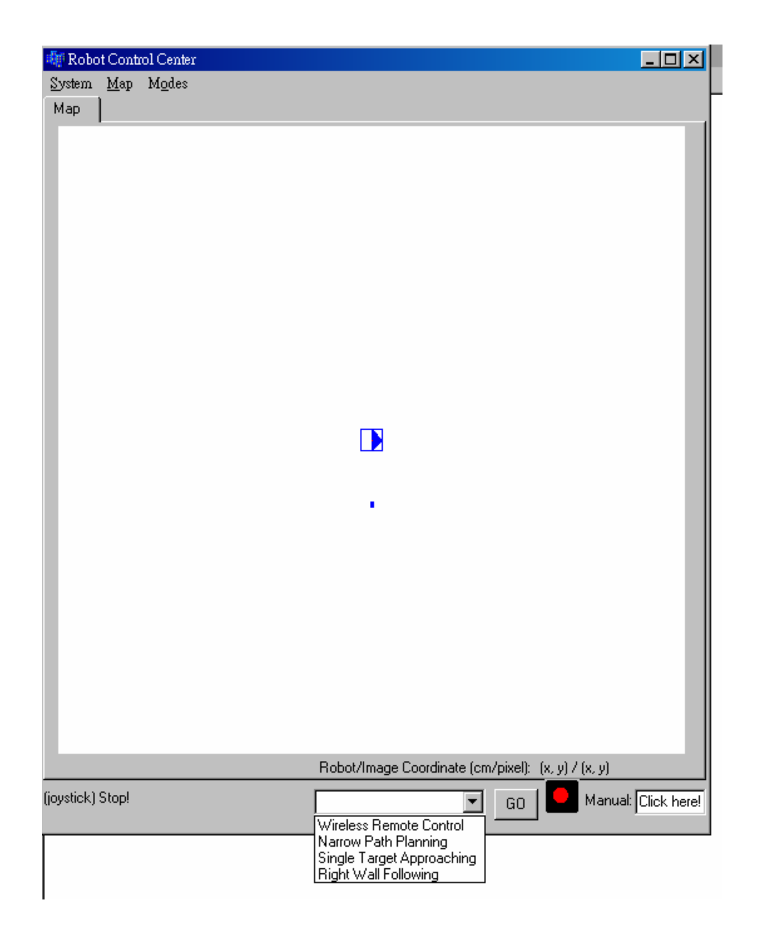
  a. 速度改為 600 mm/s。

  b. 避障距離及沿牆走距離改為 50 cm。

若搖桿擁有 Z 軸調整功能，則當 Z 軸為正時，可加快遙控速度為兩倍。不過請小心使用，遙控速度過快時，避障距離可能會不夠大而造成來不及避障。

4. 展示

目前展示功能共包含四項：Wireless Remote Control、Narrow Path Planning、Single Target Approaching、和 Right Wall Following。這些展示功能特別是為了讓使用者能夠在最短時間內看到機器人的功能，而不需要學習複雜的設定和使用方法。如下圖所示：



- Wireless Remote Control

此功能為使用遠端電腦來遙控機器人，並利用地圖與 camera 影像來幫助使用者判斷該如何遙控機器人。並且會開啟避障模式(Modes→Obstacle Avoidance→Both)，以避免由於 camera 視角不夠寬廣，導致容易撞到障礙物。請注意若要使用遠端電腦遙控機器人，需要在執行此程式前先執行遠端連線程式，確定機器人和遠端電腦已連線後再執行此功能。要執行此功能，可直接按展示功能選單並選擇 Wireless Remote Control。按下後會顯示下面的視窗：



請依照視窗中的文字指示進行此展示功能。若是要取消此功能，請按 System→Stop 或是 Ctrl+S 即可(或是在搖桿有連線時按 Button 1)。

● Narrow Path Planning

此功能為使用路徑規劃方法，讓使用者能夠僅僅設定單一目標點的情況下，能夠自動依據目前地圖上的資訊來設定出中間路徑並往目標點移動。請注意為了讓機器人能夠順利且準確地抵達目標點，請務必確定執行此展示功能時，地圖區塊中的地圖資訊為準確的，且沒有包含動態障礙物(如人)。另外環境盡量不要有太狹窄的走道，或是表面不平整的障礙物(指的是並非如牆壁般平整，而是有突出處的障礙物，如桌子、椅子等)。要執行此功能，可直接按展示功能選單並選擇 Narrow Path Planning。按下後會顯示下面的視窗：



請依照視窗中的文字指示進行此展示功能。若是要取消此功能，請按 System→Stop 或是 Ctrl+S 即可(或是在搖桿有連線時按 Button 1)。

● Single Target Approaching

此功能為單點目標點移動，讓機器人移動到一目標點上。與 Narrow Path Planning 的不同處在於，此功能是假設機器人目前位置與目標點間沒有障礙物的情況，因此會以直線方式往目標點移動。若在移動過程當中發現有障礙物擋住，則機器人會設法避開之，並在繞開障礙物過程中試圖繼續往目標點移動。要執行此功能，可直接按展示功能選單並選擇 Single Target Approaching。按下後會顯示下面的視窗：



請依照視窗中的文字指示進行此展示功能。若是要取消此功能，請按 System→Stop 或是 Ctrl+S 即可(或是在搖桿有連線時按 Button 1)。

- Right Wall Following

此功能為使機器人自主沿著右牆移動，搭配地圖繪製時，可以達到自建地圖的效果。若在過程中發現有障礙物，則會以原地左轉方式避開之。要執行此功能，可直接按展示功能選單並選擇 Right Wall Following。按下後會顯示下面的視窗：

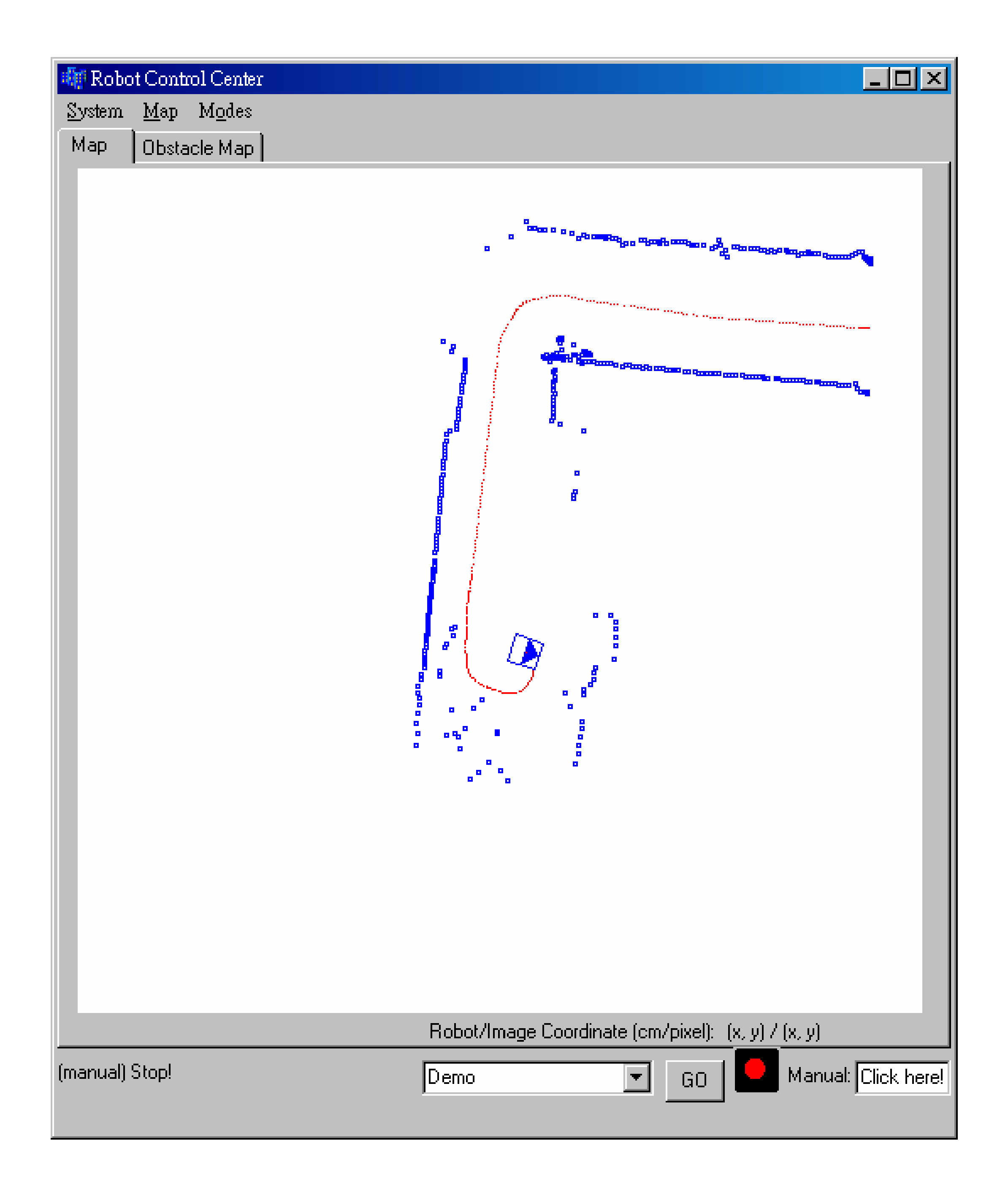


請依照視窗中的文字指示進行此展示功能。若是要取消此功能，請按 System→Stop 或是 Ctrl+S 即可(或是在搖桿有連線時按 Button 1)。

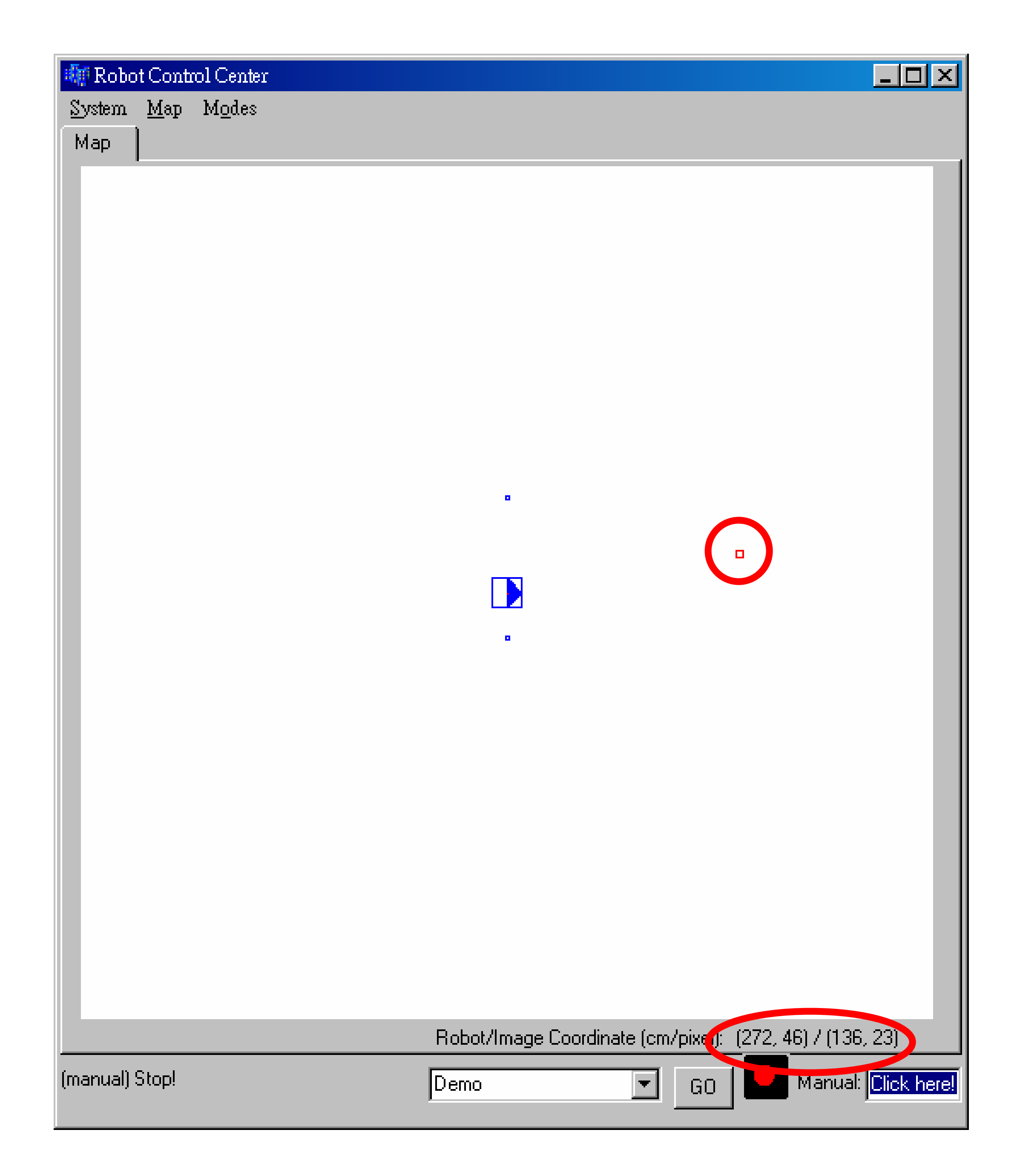
5. 地圖

地圖區塊中僅有一個頁面：Map。

1. Map



在地圖頁面中，可以顯示機器人本身的軌跡以及地圖資訊。其中，可以用滑鼠點擊地圖中的任一位置，會在 Robot/Image Coordinate 處顯示該點所代表的機器人座標位置/地圖中的位置，如下圖：

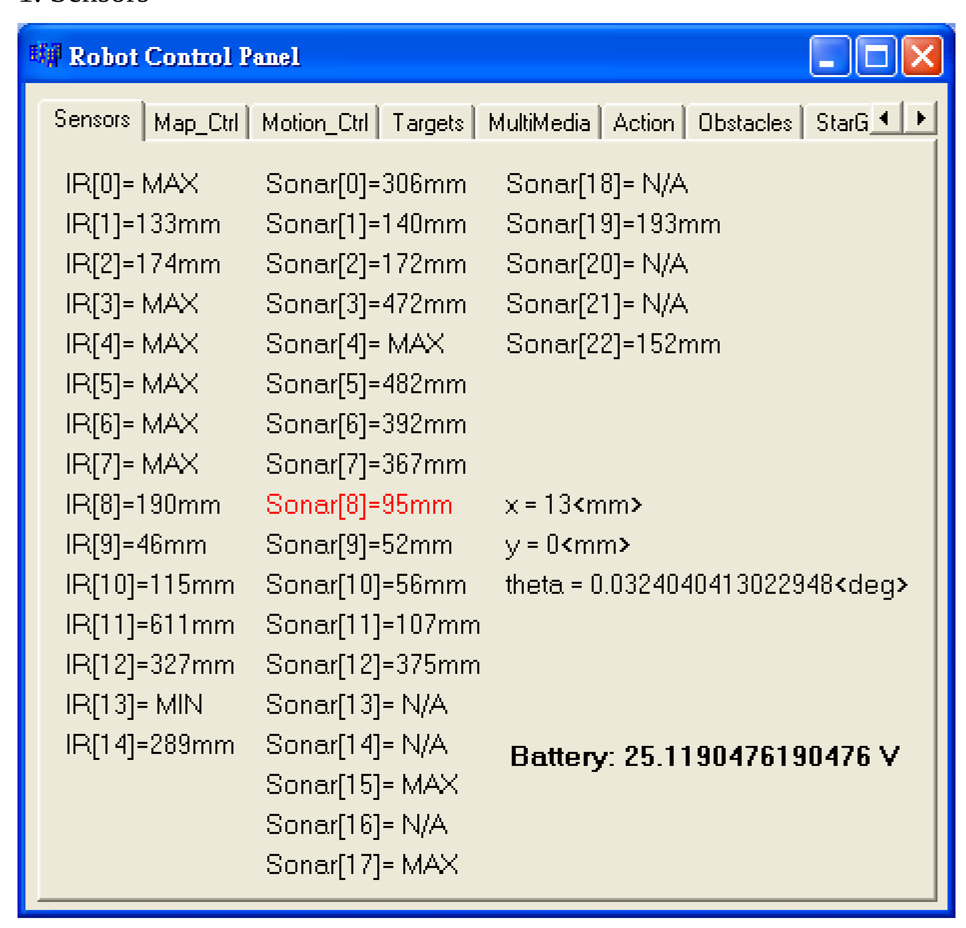


其中滑鼠點到的影像 pixel 位置是(136, 23)，但由於此時比例尺是設為 2，所以此點是對應到地圖中的(272, 46)，單位是 cm。其中地圖顯示最大範圍為 500x500，而比例尺最大為 16，所以最多可以顯示 8000x8000cm，也就是 80x80m 的地圖。

6. 資訊

資訊區塊中有九個頁面：(1) Sensors (2) Map_Ctrl (3) Motion_Ctrl (4) Targets (5) MultiMedia (6) Action (7) Obstacles (8) StarGazer (9) Test。以下分別做介紹：

1. Sensors

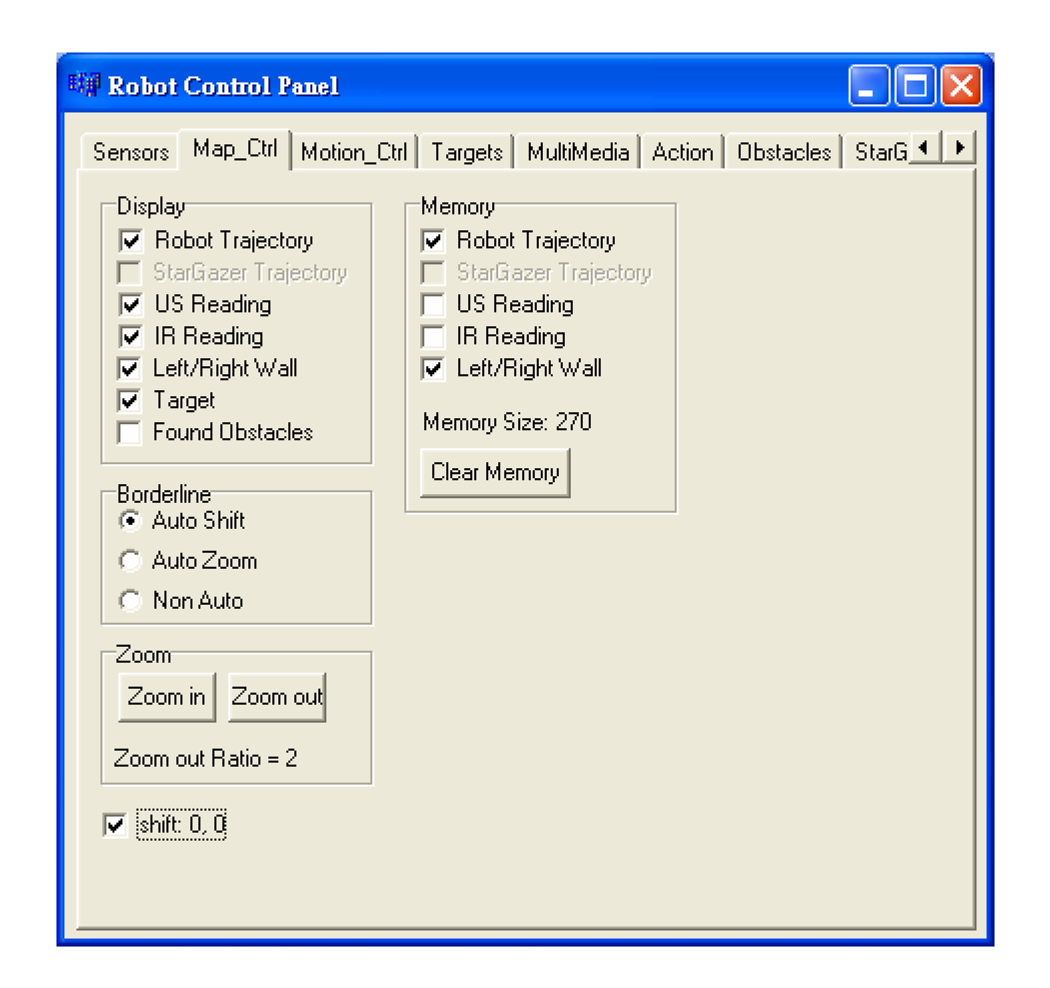


Sensors 頁面包含了以下資訊：

- 紅外線感測器讀值 (IR[0]~IR[14])
- 超音波感測器讀值 (Sonar[0]~Sonar[22])
- 機器人目前位置及朝向角(x, y, theta)
- 電池電力

 若是電池電力不足 22V，就會以紅色顯示，提醒使用者該充電了。

2. Map_Ctrl



Map_Ctrl 頁面包含了以下資訊：

- Display (地圖顯示設定)
  - Robot Trajectory

    若有打勾，則會在地圖區塊中顯示記憶體中儲存的機器人軌跡。若取消打勾，則地圖區塊中不會顯示機器人軌跡。機器人軌跡會以紅色線段顯示，同時顯示機器人目前的位置及朝向角。

  - StarGazer Trajectory

    若有打勾，則會在地圖區塊中顯示記憶體中儲存的 StarGazer 定位系統的軌跡。若取消打勾，則地圖區塊中不會顯示 StarGazer 軌跡。StarGazer 軌跡會以黑色線段顯示，顯示定位系統的位置。

  - US Reading

    若有打勾，則會在地圖區塊中顯示記憶體中儲存的超音波讀到的障礙物位置。若取消打勾，則地圖區塊中不會顯示超音波讀到的障礙物位置。超音波讀到的

障礙物位置以藍色小點顯示。即利用機器人目前的位置及朝向角，以及各超音波的讀值，和各超音波裝置在機器人上面的相對位置資訊，所推得的障礙物位置。

- IR Reading

  若有打勾，則會在地圖區塊中顯示記憶體中儲存的紅外線讀到的障礙物位置。若取消打勾，則地圖區塊中不會顯示紅外線讀到的障礙物位置。紅外線讀到的障礙物位置以藍色小點顯示。即利用機器人目前的位置及朝向角，以及各紅外線的讀值，和各紅外線裝置在機器人上面的相對位置資訊，所推得的障礙物位置。

- Left/Right Wall

  若有打勾，則會在地圖區塊中顯示記憶體中儲存的左右牆壁位置。若取消打勾，則地圖區塊中不會顯示左右牆壁位置。左右牆壁位置會以藍色粗點顯示，即利用機器人目前的位置及朝向角，以及左右側超音波、紅外線的讀值，和各左右側超音波、紅外線裝置在機器人上面的相對位置資訊，所推得的左右牆壁位置。

- Target

  若有打勾，則會在地圖區塊中顯示記憶體中儲存的目標點位置。若取消打勾，則地圖區塊中不會顯示目標點位置。已設定之目標點位置，其前十個目標點會以紅色阿拉伯數字標示其編號以及位置，其他的會以黑色正方形表示之。

- Found Obstacles

  若有打勾，且正在執行 Modes→Obstacle Avoidance→Map 或是 Modes→Obstacle Avoidance→Fusion，則會在地圖區塊中顯示被偵測到的障礙物(紅色)。若取消打勾，則地圖區塊中不會顯示被偵測到的障礙物。

- Borderline (通過邊界之設定)

  - Auto Shift

    當機器人移動超過邊界時，會自動 Shift 三分之一個視窗，以便能繼續觀察機器人軌跡及其附近的障礙物位置。

- Auto Zoom

  當機器人移動超過邊界時，會自動 Zoom out 一倍，以便能繼續觀察機器人軌跡及其附近的障礙物位置。

- Non Auto

  當機器人移動超過邊界時，不會自動調整視窗。

- Zoom (手動放大縮小設定)

  - Zoom in

    手動 Zoom in 一倍，也就是 Zoom out Ratio 變成一半。

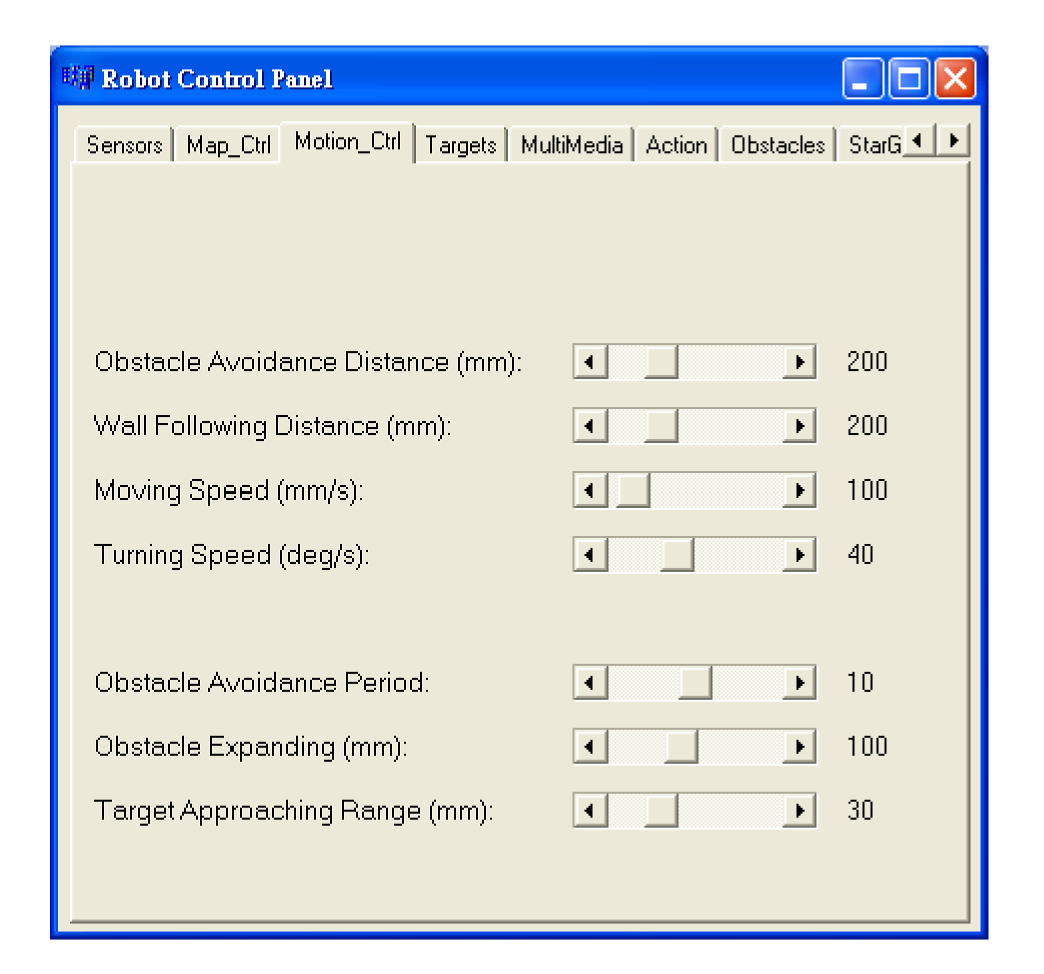
  - Zoom out

    手動 Zoom out 一倍，也就是 Zoom out Ratio 變成一倍。

  - Zoom out Ratio

    地圖 pixel 與實際座標比例尺，範圍在 1 到 16 之間。若為 1，則地圖中每個 pixel 代表 1cm；若為 16，則地圖中每個 pixel 代表 16cm。

3. Motion_Ctrl



Motion_Ctrl 頁面包含了以下資訊：

- Obstacle Avoidance Distance (mm)

  也就是障礙物離機器人的距離要多遠會啟動避障功能。

- Wall Following Distance (mm)

  決定機器人應該要沿牆壁(左或右)多少距離來移動。

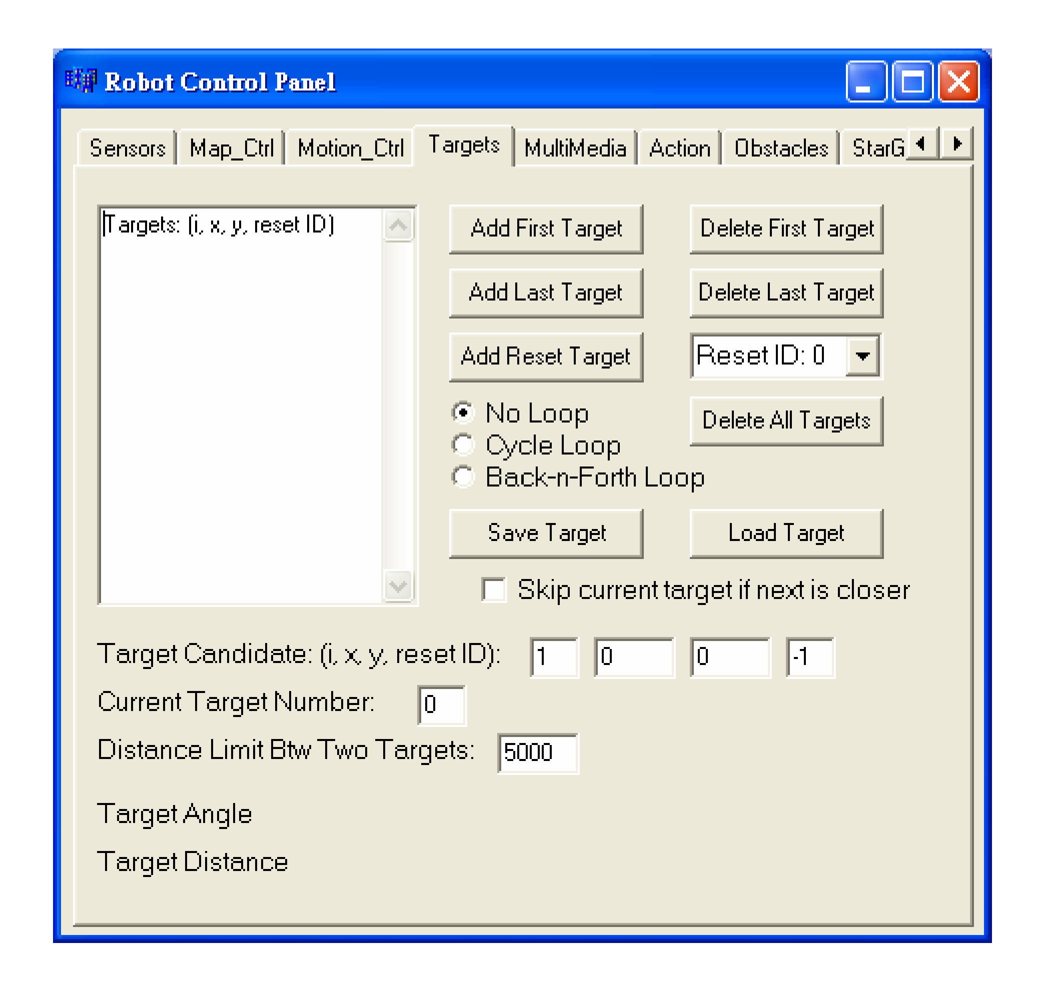
- Moving Speed (mm/s)

  機器人前進後退的線性速度。

- Turning Speed (deg/s)

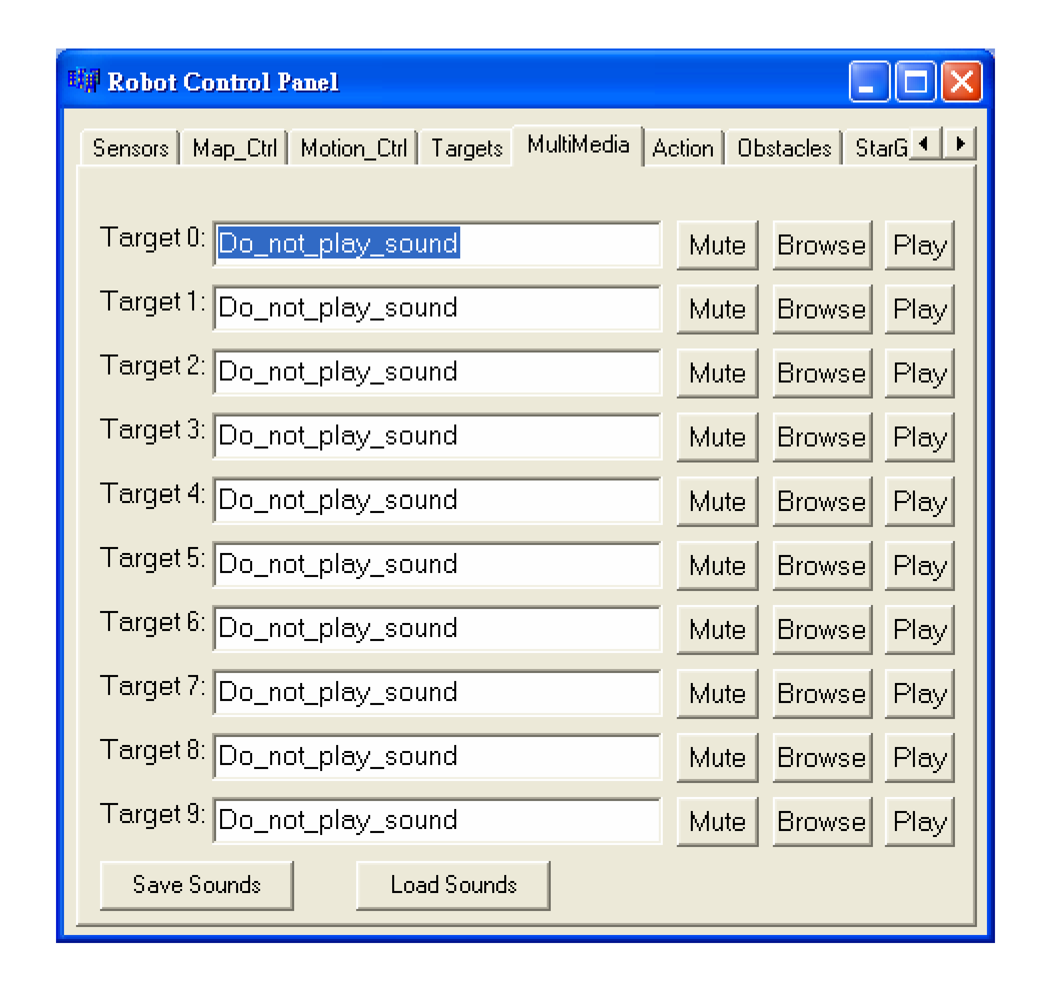
  機器人原地左右轉的旋轉速度。

4. Targets



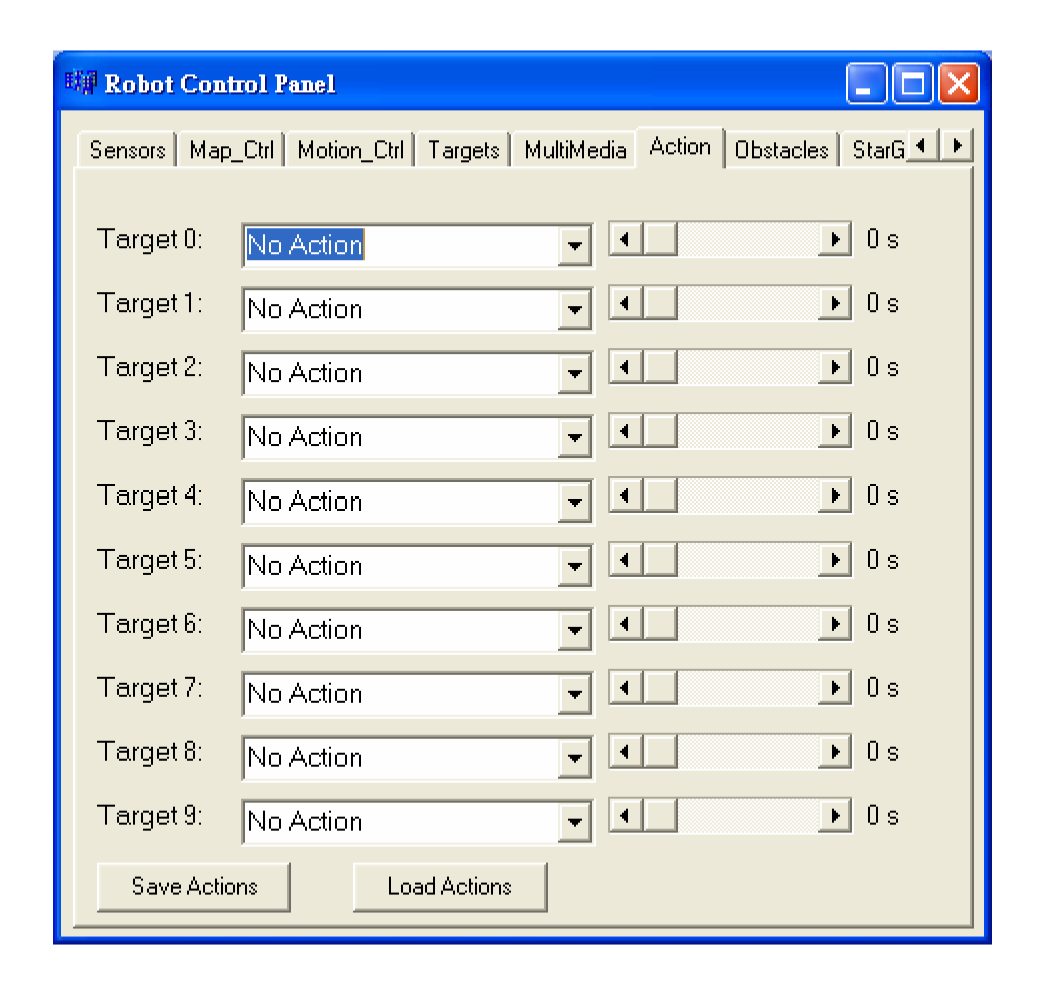
Targets 頁面所包含的資訊請詳見選單區塊的 Modes→Target Approaching。

5. MultiMedia



MultiMedia 頁面可以設定當機器人抵達前十個目標點時，各別會發出的音檔或是影像檔。音檔的副檔名為.wav，影像檔的副檔名為.avi。按下 Browse 按鈕可以選擇影音檔，按下 Play 按鈕可以試播該影音檔。若是按下 Mute 按鈕，則抵達該目標點時不會播放任何影音檔。

6. Action

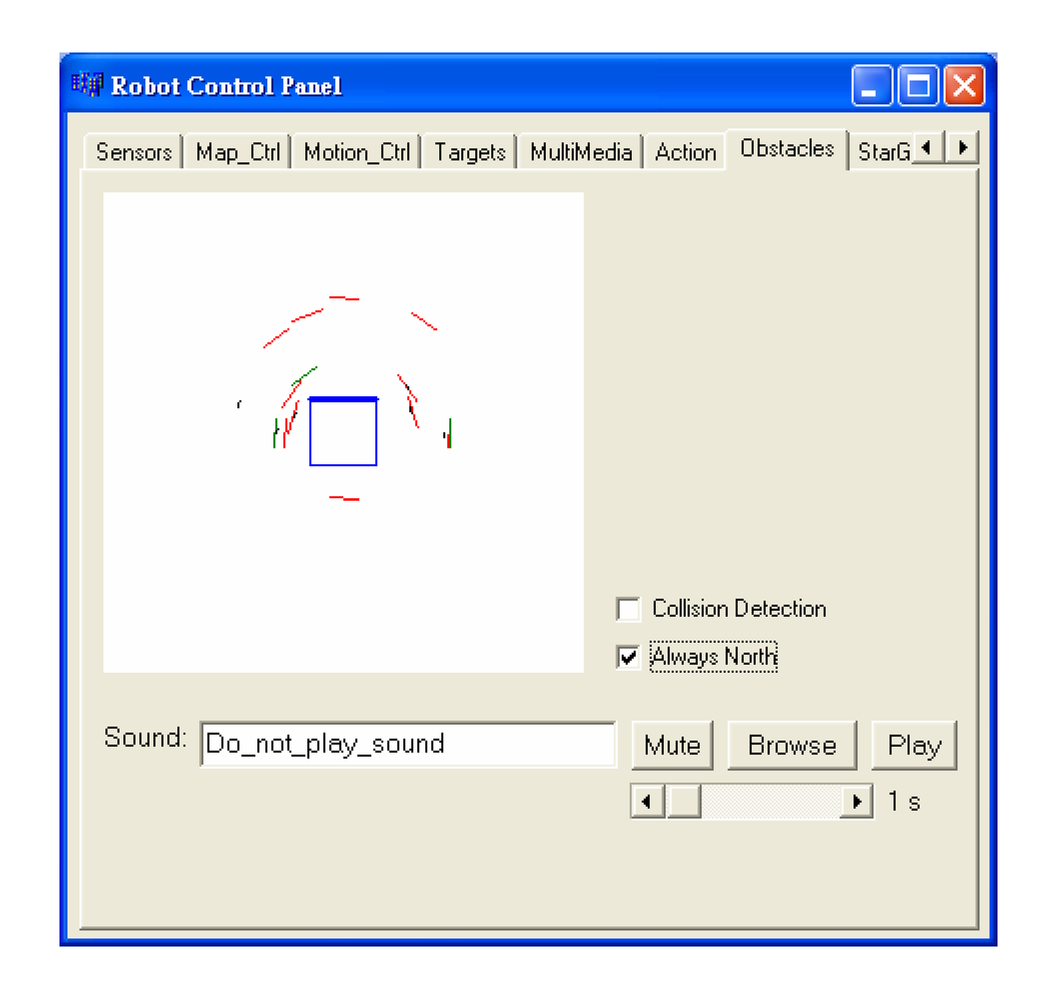


Action 頁面可以設定當機器人抵達前十個目標點時，各別會執行的動作。說明如下：

- No Action
  不做任何動作。
- Turn Left 90 degrees
  向左原地旋轉 90 度。
- Turn Right 90 degrees
  向右原地旋轉 90 度。
- Turn 180 degrees
  向左原地旋轉 180 度。

另外，可以設定抵達各目標點時，應該要停止的時間。也就是停止若干秒之後，才繼續往下一個目標點前進。此時間可以搭配 MultiMedia 頁面的影音檔長度來設定。

7. Obstacles



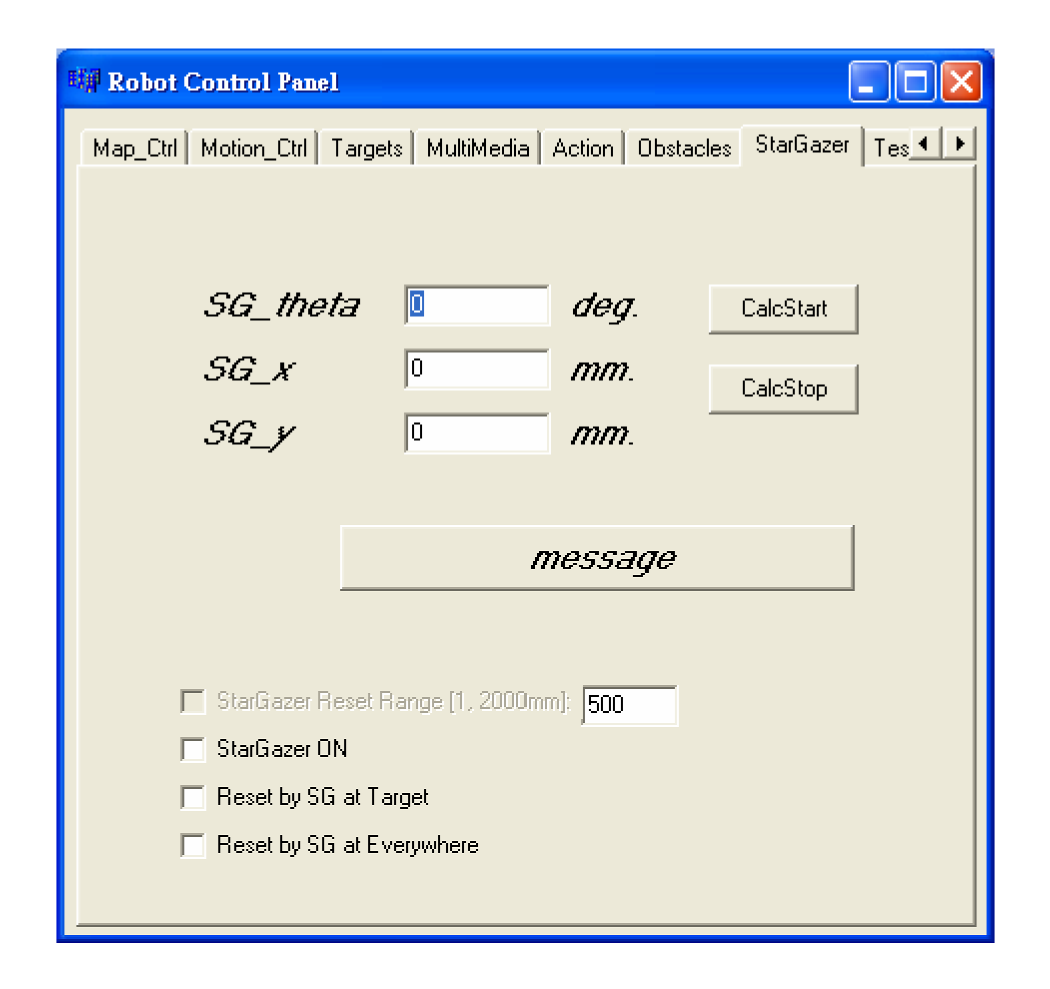
Obstacles 頁面顯示超音波與紅外線所偵測到的障礙物與機器人的相對位置圖。詳細說明請參見選單區塊的 Map→Show Obstacles。

此外，若是勾選 Collision Detection 時，機器人會自動偵測是否有撞擊到障礙物。若是偵測到機器人已撞擊到障礙物，則會播放警示喇叭音，同時顯示以下視窗。



另外，也可以設定看到障礙物時，應播放的音效以及播放時間。設定方法請參考 MultiMedia 以及 Action 頁面。

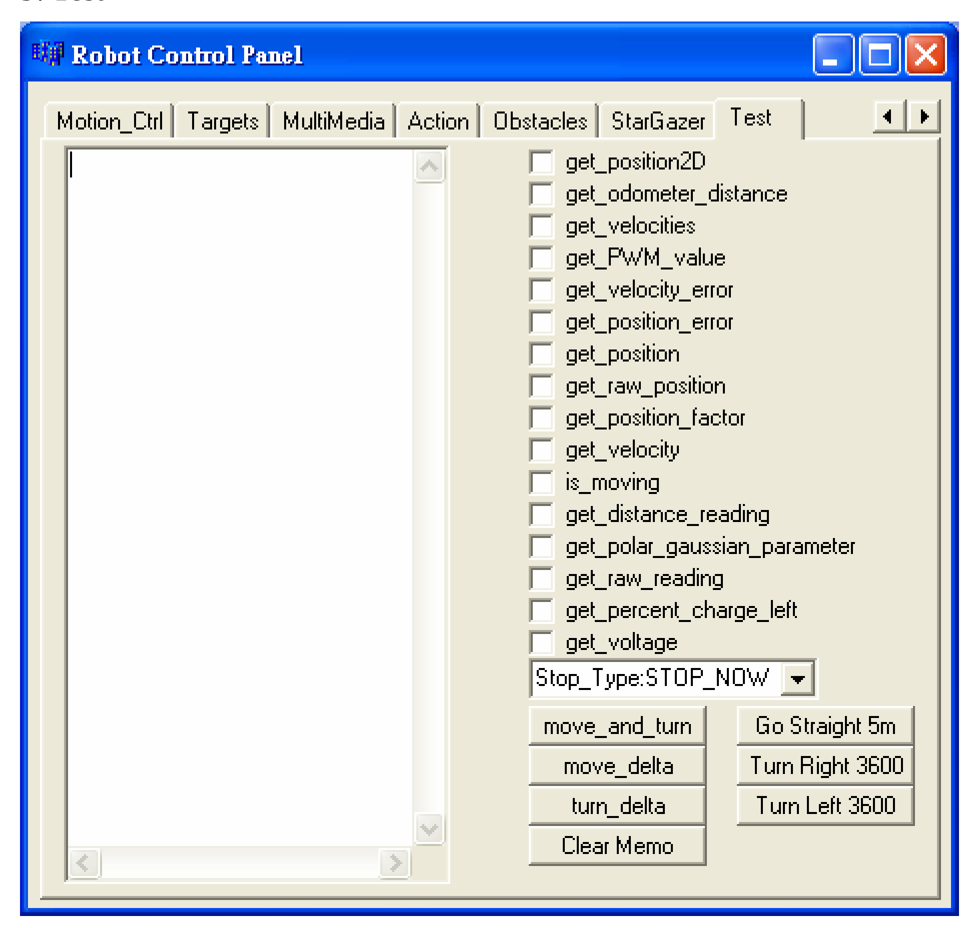
8. StarGazer



StarGazer 頁面顯示使用 StarGazer 定位系統所需的設定和按鈕。其中 SG_theta、SG_x 和 SG_y 分別代表 StarGazer 定位系統中的角度、x 座標和 y 座標。其他說明如下：

- CalcStart
  開始執行 StarGazer 定位功能
- CalcStop
  關閉 StarGazer 定位功能
- Message
  顯示 StarGazer 做 Reset 的相關訊息
- StarGazer Reset Range [1, 2000mm]
  一旦打勾此項，則每次做座標 Reset 時都會比對機器人座標和 StarGazer 座標，若差異超過此值，就不執行座標 Reset。
- StarGazer ON
  開啟 StarGazer
- Reset by SG at Target
  開啟 StarGazer Reset 機器人座標系統功能。每當機器人抵達任意一個目標點時就會執行座標 Reset。
- Reset by SG at Everywhere
  開啟 StarGazer Reset 機器人座標系統功能。每當 StarGazer 讀到一次正確定位資訊時就會執行座標 Reset。

9. Test



Test 頁面包含了所有 HAL 所提供的控制機器人功能。勾選任何一個功能，都會將現在機器人狀態顯示在左邊的視窗中。功能說明如下：

- get_position2D
  顯示機器人中心位置。
- get_odometer_distance
  顯示機器人中心離原點的距離。
- get_velocities
  顯示機器人線性速度及角速度。
- get_PWM_value
  顯示馬達 PWM 百分比。
- get_velocity_error
  顯示左右輪速度誤差。
- get_position_error
  顯示左右輪位置誤差。
- get_position
  顯示左右輪離座標系統原點的距離。

- get_raw_position
  顯示左右輪位置原始值。
- get_position_factor
  顯示位置原始值與實際值的比例。
- get_velocity
  顯示左右輪速度。
- is_moving
  顯示機器人是否正在移動。
- get_distance_reading
  顯示目前所有感測器的讀值。
- get_polar_gaussian_parameter
  此為備用功能。
- get_raw_reading
  顯示目前所有感測器的原始值。
- get_percent_charge_left
  顯示電池所剩的電力。
- get_voltage
  顯示電池目前伏特數。

另外，以下是移動方面的功能：

- Stop_Types
  可選擇機器人停止的方式，有以下三種：
  - STOP_NOW
    用控制器將馬達直接鎖死，是最強力的煞停方式。
  - SOFT_STOP
    用橋式電路將馬達以內迴路方式做煞停。
  - RELEASE
    把橋式電路全部打開，類似放空檔，僅靠地面摩擦力煞停，是最弱的煞停方式。
- move_and_turn
  會依序執行以 10 公分/秒速度前進、以 10 公分/秒速度後退、以 40 度/秒速度順時針旋轉、以 40 度/秒速度逆時針旋轉。最後會以 Stop_Types 所選擇的方式煞停。
- move_delta
  會依序執行以 10 公分/秒速度前進 10 公分、以 10 公分/秒速度後退 10 公分。最後會以 Stop_Types 所選擇的方式煞停。
- turn_delta
  會依序執行以 40 度/秒速度順時針旋轉 90 度、以 40 度/秒速度逆時針旋轉 90 度。最後會以 Stop_Types 所選擇的方式煞停。

按下 Clear Memo 可將左邊視窗清空。

另外 Go Straight 5m、Turn Right 3600 和 Turn Left 3600 等三個按鈕，分別會執行往前走 5 公尺、原地往右轉十圈和原地往左轉十圈等三項功能。在執行前請注意環境是否夠寬敞。

7. 結論

在本 Demo 程式操作手冊中，提供了展示 MR-W1 功能的導航技術使用說明，包含選單區塊、訊息與遙控區塊、地圖區塊、以及資訊區塊等。此導航技術可以執行顯示資訊、導航參數調整、繪製地圖、地圖資訊存取、遠端遙控、障礙物分布偵測、避障、功能測試、遠端監控、以及自主巡航等功能。

MicroStar International
iHOMER
Intelligent Indoor Robotic Toolkits

機器人高臺超音波操作手冊

說明書

Version 0G



Robot Team

Research Center @ MSI

3 JAN, 2008

超音波感測器

iHOMER 機器人之標準配備是 11 顆超音波感測器，最多可裝置 23 顆超音波感測器，用於障礙物偵測、避開動態或靜態障礙物、特徵識別、定位、以及沿牆行走、自繪地圖等導航等功能。每兩顆超音波間隔 18 度。提供 1~50 公分的偵測距離。其中底層最多配置了 12 顆超音波(有 1 顆放置在後方當倒車雷達)，而上層最多配置了 11 顆超音波。圖 3.1 以及圖 3.2 為以 PRO/E 顯示每個超音波位置與座標之關係，其編號由 1~23，而在圖 3.3 顯示超音波測試軟體所進行的實體測試，因為 C 語言關係，其實際編號少 1。

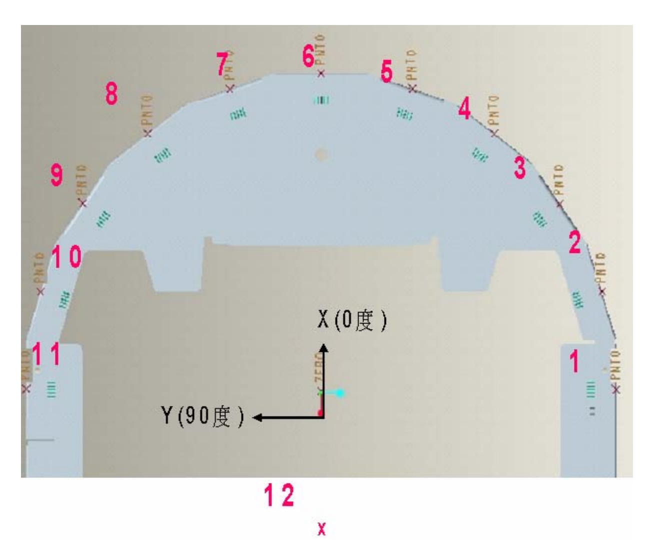


圖 3.1：底層超音波配置與編號

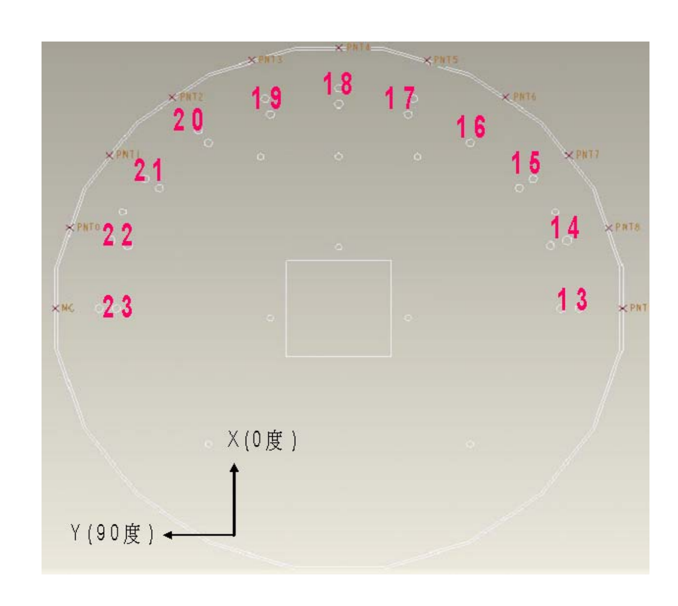


圖 3.2：上層超音波配置與編號

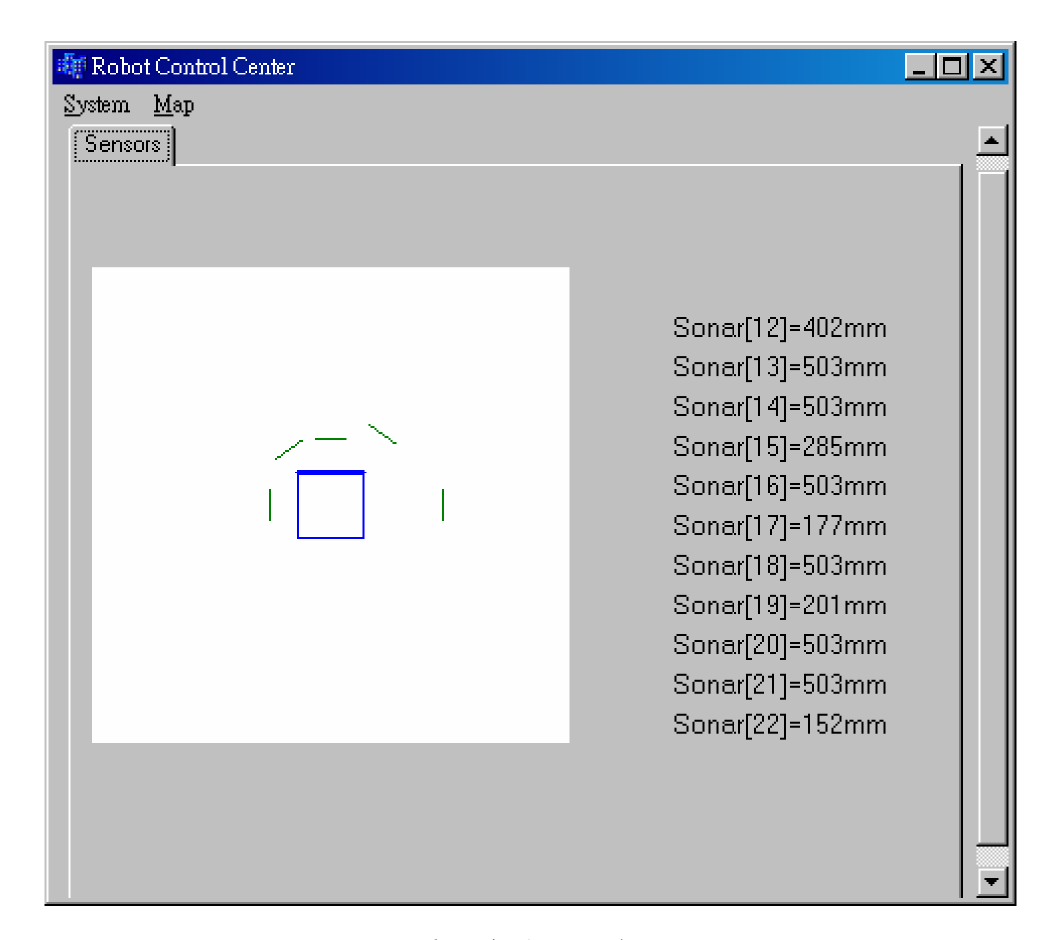


圖 3.3：上層超音波配置與測試 Demo

請注意以安全的方式操作機器人，避免在潮濕地面、樓梯附近、地面崎嶇不平、…等不安全的環境下，以目視或遠端控制方式操控機器人，將可以確保使用人與機器人的性能與安全，有任何問題請連繫微星科技機器人事業之業務或工程師 (02)32345599#1039 October，進行疑難排除。





Introduction

This software is used to provide necessary functionalities for the chassis of ICCR. It doesn’t include the Embedded system software which will be implemented separately.

Outline

- Application interface
  - Motor command
    - move_and_turn()
    - move_delta()
    - turn_delta()
    - stop()
    - set_move_power()
    - set_idle_power()
  - Odometer
    - get_position2D()
    - get_odometer_distance()
    - get_velocities()
    - reset_odometer()
  - Motor controller
    - get_PWM_value()
    - get_velcoity_error()
    - get_position_error()
  - Motor query
    - get_position()
    - get_raw_position()
    - get_position_factor()
    - get_velocity()
    - is_moving()
  - Range sensor
    - get_distance_reading()



            - get_polar_gaussian_parameters()
            - get_raw_reading()
        - Battery
            - get_percent_charge_left()
            - get_voltage()
            - get_charge_state()
- Configuration files
    - Sonar configuration file
    - IR configuration file
    - Communication configuration file
    - Power configuration file
    - Drive configuration file

Class reference

IMotorCommand Interface Reference

This class provides a resource interface to control a motor device. The priority of the member functions are: set_move_power = set_idle_power > move_delta = turn_delta > move_and_turn.

Namespaces

ICCR

Public Member Functions

Result move_and_turn (TicketId ticket, double velocity, double acceleration, double angular_velocity, double angular_acceleration)

ticket [in];	The (TicketId) for security purposes.	
velocity [in];	The desired velocity	<units : mm/s>
acceleration [in];	The desired acceleration in order to satisfy the desired velocity	<units : mm/s^2>
angular_velocity [in];	The desired angular velocity	<units : rad/s>
angular_acceleration [in];	The desired angular acceleration in order to satisfy the desired	

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



velocity <units : rad/s^2>

Result RESULT_INVALID_ARGUMENT,
RESULT_OUT_OF_BOUND,
RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

This function is to control the movement of a robot.

Result	**Description**
RESULT_INVALID_ARGUMENT	The input argument is not reasonable
RESULT_OUT_OF_BOUND	The input argument is over the hardware spec
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result move_delta (TicketId ticket, IResourceCallback callback, CallbackId callback_id, double displacement, double velocity, double acceleration)

ticket [in]; The (TicketId) for security purposes.
callback [in]; Callback function to use to verify command completion
callback_id[in]; Reserved.
displacement [in]; Desired displacement distance <units : mm>
velocity [in]; Desired velocity <units : mm/s>
acceleration [in]; Desired acceleration <units : mm/^2>

Result RESULT_INVALID_ARGUMENT,
RESULT_OUT_OF_BOUND,
RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

- This function is to control the robot to move forward/backward several distance (unit: mm). The callback function will be called after the robot has finished this command.

Result	**Description**
RESULT_INVALID_ARGUMENT	The input argument is not reasonable
RESULT_OUT_OF_BOUND	The input argument is over the hardware spec
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented



Result turn_delta (TicketId ticket, IResourceCallback callback, CallbackId callback_id, double angular_displacement, double angular_velocity, double angular_acceleration)

ticket [in];	The (TicketId) for security purposes.	
callback [in];	The callback function used to test command completion	
callback_id[in];	Reserved.	
angular_displacement [in];	Desired angular heading	<units : rad>
angular velocity [in];	Desired angular velocity.	<units : rad/s>
angular acceleration [in];	Desired angular acceleration.	<units : rad/s^2>

Result RESULT_INVALID_ARGUMENT,
RESULT_OUT_OF_BOUND,
RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

- This function is to control the robot to rotate clockwise/counterclockwise (unit: rad). The callback function will be called after the robot has finished this command.

Result	**Description**
RESULT_INVALID_ARGUMENT	The input argument is not reasonable
RESULT_OUT_OF_BOUND	The input argument is over the hardware spec
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result stop (TicketId ticket, IResourceCallback callback, CallbackId callback_id, StopType stop_type)

ticket [in];	The (TicketId) for security purposes.
callback [in];	The callback function used to test command completion.
callback_id [in];	Reserved.
stop_type [in];	Specific manner used for stopping.

Result RESULT_INVALID_STATE,
RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:



- This function is to stop the robot by several manners. The callback function will be called after the robot has finished this command.

Result	**Description**
RESULT_INVALID_STATE	The input state is not existed
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result set_move_power (TicketId ticket, IResourceCallback callback, CallbackId callback_id, double move_power)

ticket [in]; The (TicketId) for security purposes.
callback [in]; The callback function used to test command completion.
callback_id[in]; Reserved.
move_power [in]; Percentage of power used for movement

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

- This function is to turn on the system power. If move_power is set to 0, the system power will be shut down. If the hardware does not support this function, move_power should be set to NULL.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result set_idle_power (TicketId ticket, IResourceCallback callback, CallbackId callback_id, double idle_power)

ticket [in]; The (TicketId) for security purposes.



callback [in]; The callback function used to test command completion.
callback_id[in]; Reserved.
idle_power [in]; Percentage of power used during idling

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

- This function is to set the system power during idling. If the hardware does not support this function, idle_power should be set to NULL.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

IOdometry Interface Reference

This class defines a resource interface to an odometry driver which converts raw motor encoder values to x,y,theta position using motor properties, calibrations, and possibly error correction.

Namespaces

ICCR

Public Member Functions

Result get_position2D(TicketId ticket, Pose2DT *position);

ticket [in]; The (TicketId) for security purposes.
position [out]; Returns current position <unit mm, rad, ms>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

The get position function obtains a calibrated position reading with no error estimation (for now).



Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_odometer_distance(TicketId ticket, double *position)

ticket [in]; The (TicketId) for security purposes.
position [out]; Returns current position <unit mm>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

The get odometer distance function obtains the distance traversed since the last odometer reset.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_velocities (TicketId ticket, double *v, double *w)

ticket [in]; The (TicketId) for security purposes.
v [out]; Returns the velocity <unit mm/s>
w [out]; Return the angular velocity <unit rad/s>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

The get_velocities function gets the current velocity and angular velocity.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result reset_odometer(TicketId ticket)

ticket [in]; The (TicketId) for security purposes.



Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
Reset odometer x, y, theta to zero.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result reset_odometer(TicketId ticket, Pose2DT *position)
ticket [in]; The (TicketId) for security purposes.
position[in]; The position and direction you want to reset.

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
Reset odometer x, y, theta to the desired values.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

IController Interface Reference

This class provides an interface to query controller information.

Namespaces

ICCR

Public Member Functions

Result get_PWM_value(TicketId ticket, double *percentage);
ticket [in]; The (TicketId) for security purposes.
percentage [out]; The current PWM value <unit : %>



Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
The get_PWM_value function can obtain the left/right motor controlling energy.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_velcoity_error(TicketId ticket, double *error);
ticket [in]; The (TicketId) for security purposes.
error [out]; The error between the desired and actual velocity of one motor.
< units : mm/s >

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
This function can obtain the error between the desired and actual velocity of one motor.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_position_error(TicketId ticket, double *error);
ticket [in]; The (TicketId) for security purposes.
error [out]; The error between the desired and actual position of one motor.
< units : mm >

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
This function can obtain the error between the desired and actual position of one motor.

Result	**Description**



RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

IMotorQuery Interface Reference

The IMotorQuery class provides a resource interface for queries of a motor device.

Namespaces

ICCR

Public Member Functions

Result get_position (TicketId ticket, double *position)

ticket [in]; The (TicketId) for security purposes.

position [out]; Returns current position <unit mm>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

The get_position function returns the current position of one motor device. It is derived from raw_position() and position_factor().

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_raw_position (TicketId ticket, double *raw_position)

ticket [in]; The (TicketId) for security purposes.

raw position [out]; Returns current position <unit: encoder counts>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.



Description:

The get_raw_position function returns the current raw position of a motor device.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_position_factor (TicketId ticket, double *position_factor)

ticket [in]; The (TicketId) for security purposes.

position_factor [out]; Factor used to convert raw position to mm

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

This function can get the factor written in the configuration file for converting the raw position to mm.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_velocity (TicketId ticket, double *velocity)

ticket [in]; The (TicketId) for security purposes.

velocity [out]; Current motor velocity <unit mm/s>

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

This function returns the current motor velocity.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result is_moving (TicketId ticket, bool *moving)



ticket [in]; The (TicketId) for security purposes.
moving [out]; True if the robot is moving, False otherwise

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
This function is to check if the robot is moving or not.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

IRangeSensor Interface Reference

The IRangeSensor class provides a resource interface to a range sensor, that is, a sensor that returns a distance reading to a sensed physical object.

Namespaces

ICCR

Public Member Functions

Result get_distance_reading (TicketId ticket, Timestamp _timestamp, double *distance)
ticket [in]; The (TicketId) for security purposes.
timestamp [out]; The timestamp after reading the device
distance [out]; The returned distance reading <unit mm>
Result RESULT_OVER_BOUND,
RESULT_UNDER_BOUND,
RESULT_SUCCESS,
RESULT_INVALID_TYPE.
Description:
This function is to obtain the distance reading (mm) of a specific sensor

Result	Description
RESULT_OVER_BOUND	The sensed object is over the measurable range



RESULT_UNDER_BOUND	The sensed object is under the measurable range
RESULT_SUCCESS	The function is successfully implemented
RESULT_INVALID_TYPE	The input ticket ID is invalid

Result get_raw_reading (TicketId ticket, Timestamp _timestamp, double *raw reading)

ticket [in]; The (TicketId) for security purposes.

timestamp [out]; The timestamp of the reading

raw reading [out]; The returned distance reading

Result RESULT_OVER_BOUND,
RESULT_UNDER_BOUND,
RESULT_SUCCESS,
RESULT_INVALID_TYPE.

Description:

This function is to obtain the distance reading (no unit) of a specific sensor

Result	**Description**
RESULT_OVER_BOUND	The sensed object is over the measurable range
RESULT_UNDER_BOUND	The sensed object is under the measurable range
RESULT_SUCCESS	The function is successfully implemented
RESULT_INVALID_TYPE	The input ticket ID is invalid

IBattery Interface Reference

This class provides a resource interface to a battery.

Namespaces

ICCR

Public Member Functions

Result get_voltage(TicketId ticket, Timestamp *timestamp, double *voltage);

ticket [in]; The (TicketId) for security purposes.

timestamp[in]; The timestamp of the reading.

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)

voltage [out]; The battery voltage <unit : V>

Result RESULT_INVALID_TYPE,
RESULT_OVER_BOUND,
RESULT_UNDER_BOUND,
RESULT_SUCCESS.

Description:
The get_voltage function can obtain the battery voltage.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_OVER_BOUND	The sensed object is over the measurable range
RESULT_UNDER_BOUND	The sensed object is under the measurable range
RESULT_SUCCESS	The function is successfully implemented

Result get_percent_charge_left(TicketId ticket, Timestamp *timestamp, double *percent_charge);

ticket [in]; The (TicketId) for security purposes.
timestamp [in]; The timestamp of the reading.
percent_charge [out]; The current charge level of the battery as a percentage of total charge.

Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:
The get_voltage function can obtain the battery voltage.

Result	**Description**
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

Result get_charge_state(TicketId ticket, bool *charging);

ticket [in]; The (TicketId) for security purposes.
charging [out]; The chassis is on charge station or not <TRUE still charging>.



Result RESULT_INVALID_TYPE,
RESULT_SUCCESS.

Description:

Check if chassis is on charge station and still charging or not.

Result	Description
RESULT_INVALID_TYPE	The input ticket ID is invalid
RESULT_SUCCESS	The function is successfully implemented

IChassisConf Interface Reference

This class provides the chassis configuration information.

Namespaces

ICCR

Public Member

static CHASSIS_RESOURCE::IComm *Comm;

Comm configuration data structure:

- ID
- CommPort
- BaudRate
- StopBit
- ByteSize

static CHASSIS_RESOURCE::IRangeSensor *IR;

IR sensor configuration data structure:

- ID
- Manufacturer
- Model
- Type
- Posture
  - x

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



- y
- z
- roll
- pitch
- yaw

- Range
  - MaxDistance
  - MinDistance
  - Radius

static CHASSIS_RESOURCE::IRangeSensor *Sonar;

Sonar sensor configuration data structure:

- ID
- Manufacturer
- Model
- Type
- Posture
  - x
  - y
  - z
  - roll
  - pitch
  - yaw
- Range
  - MaxDistance
  - MinDistance
  - Radius

static CHASSIS_RESOURCE::IRangeSensor *Bumper;

Bumper sensor configuration data structure:

- ID
- Manufacturer
- Model
- Type
- Posture
  - x



- y
- z
- roll
- pitch
- yaw

- Range
  - MaxDistance
  - MinDistance
  - Radius

static CHASSIS_RESOURCE::IDriver *Driver;

Driver configuration data structure.

- ID
- LeftDriver
  - ID
  - Posture
    - x
    - y
    - z
    - roll
    - pitch
    - yaw
  - Motor
    - Manufacturer
    - Model
    - Type
    - StallTorque
    - MaxSpeed
  - GearBox
    - Manufacturer
    - Model
    - Type
    - Stage
    - GearRatio
  - Encoder
    - Manufacturer



        - Model
        - Type
        - MaxBoundsAngle
        - MinBoundsAngle
        - ValuePerCircle
    - MotiveAxis
        - StallTorque
        - MaxSpeed
        - ValuePerCircle
    - Tire
        - Manufacturer
        - Model
        - Width
        - Circumference
        - Radius
    - MotorDriver
- RightDriver
    - ID
    - Posture
        - x
        - y
        - z
        - roll
        - pitch
        - yaw
    - Motor
        - Manufacturer
        - Model
        - Type
        - StallTorque
        - MaxSpeed
    - GearBox
        - Manufacturer
        - Model
        - Type
        - Stage
        - GearRatio



- Encoder
  - Manufacturer
  - Model
  - Type
  - MaxBoundsAngle
  - MinBoundsAngle
  - ValuePerCircle
- MotiveAxis
  - StallTorque
  - MaxSpeed
  - ValuePerCircle
- Tire
  - Manufacturer
  - Model
  - Width
  - Circumference
  - Radius
- MotorDriver

```
static CHASSIS_RESOURCE::IPower *Power;
```

Power configuration data structure:

- ID
- Battery
  - ID
  - Manufacturer
  - Model
  - MaxCurrent
  - MaxVoltage
  - MinVoltage
- Charger
  - ID
  - Manufacturer
  - Model
  - MaxCurrent
  - MaxVoltage

```
static unsigned int CommNum;
```



COM port number which is needed for communication between the chassis and PC.

static unsigned int IRNum;

The installed IR sensor number

static unsigned int SonarNum;

The installed Sonar sensor number

static unsigned int BumperNum;

The installed Bumper sensor number

static unsigned int DriveNum;

The installed Drive number

static unsigned int PowerNum;

The installed Power system number

ILocalizationSys Interface Reference

- This class provides an interface of StarGazer controller information.
- Communicate between MR-W1 and the StarGazer™.
- Provided a standard application interface to write commands to a StarGazer™ and to read data from a StarGazer™.
- Please refer to StarGazer User Manual for detailed explanation about all the related terms.

Namespaces

ICCR

Public Member Functions

① Result get_firmware_version(string *data);



data [out]; The StarGazer's firmware version.

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:
This function can obtain the StarGazer's firmware version.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

② Result get_ThrMode(string *data);
data [out]; Threshold mode(auto/manual).

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:
This function can obtain the StarGazer's threshold mode.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

③ Result get_MarkType(string *data);
data [out]; Mark Type(Home/Office).

Result RESULT_BUSY,
RESULT_TIMEOUT,



RESULT_SUCCESS.

Description:

This function can obtain the StarGazer's mark type.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

④ Result get_MarkMode (string *data);

data [out]; Landmark Mode(Alone/Map).

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the StarGazer's landmark mode.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

⑤ Result get_ModeSetting(string *data);

data [out]; Map building mode(Start/Stop).

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the StarGazer's map building mode.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.



RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

⑥ Result get_ID_TotalNum(unsigned short int *data);

data [out]; Number of ID(0-31,0-4095).

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the count of landmark ID.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

⑦ Result get_RefID(unsigned int *data);

data [out]; StarGazer's reference ID.

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the StarGazer's reference ID.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

⑧ Result get_MarkHeight(unsigned int *data);

data [out]; The height of landmark. <unit : mm>

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the height of landmark.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

⑨ Result get_StarGazerID(unsigned int *data);

data [out]; The present ID number.

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the present ID number.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented

⑩ Result get_ThrVal(unsigned char *data);

data [out]; StarGazer’s threshold value (0~255).

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the StarGazer’s threshold value.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.



RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

11. Result get_position(TicketId ticket, Pose2DT *data);

data.x [out]; Coordinate x-axis value (cm)
data.y [out]; Coordinate y-axis value (cm)
data.theta[out]; Value of angle (rad)

Result RESULT_BUSY,
RESULT_TIMEOUT,
RESULT_SUCCESS.

Description:

This function can obtain the StarGazer's Localization information.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_TIMEOUT	No data in.
RESULT_SUCCESS	The function is successfully implemented.

12. Result set_ThrVal(const unsigned char data);

data [in]; The StarGazer's threshold value(0~255).

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can set the StarGazer's threshold value, and the updating process needs 3~5 seconds.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.



RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

13. Result set_ID_TotalNum(const unsigned short int data);

data [in]; The count of landmark (0~31,0~4095).

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can set the count of landmark, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented

14. Result set_RefID(const unsigned int data);

data [in]; The reference ID(2~30294).

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can set the reference ID, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.



RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

15. Result set_MarkHeight(const unsigned int data);

data [in]; The height of landmark <unit : mm>

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can set the height of landmark, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

16. Result set_MarkTypeHome( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can change the mark type into home type, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.



RESULT_SUCCESS	The function is successfully implemented.

17. Result set_MarkTypeOffice( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can change the mark type into office type, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

18. Result set_MarkModeAlone( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can change the mark mode into alone type, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.



19. Result set_MarkModeMap( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:
This function can change the mark mode into map type, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

20. Result set_ThrModeManual( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:
This function can change the threshold mode into manual type, and the updating process needs 3~5 seconds.

Result	**Description**
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

21. Result set_ThrModeAuto( );



Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can change the threshold mode into auto type, and the updating process needs 3~5 seconds.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

22. Result set_CalcStart( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can start StarGazer right away.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

23. Result set_CalcStop( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,



RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:
This function can stop StarGazer right away.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

24. Result build_MapStart( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:
This function can start the process of building map.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

25. Result build_MapStop( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:



This function can stop the process of building map.

Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.

26. Result auto_HeightCalc( );

Result RESULT_BUSY,
RESULT_SYSTEM_ERROR,
RESULT_IO_ERROR,
RESULT_SUCCESS.

Description:

This function can calculate the distance between a StarGazer and a landmark automatically, and the calculating process needs 10~12 seconds.

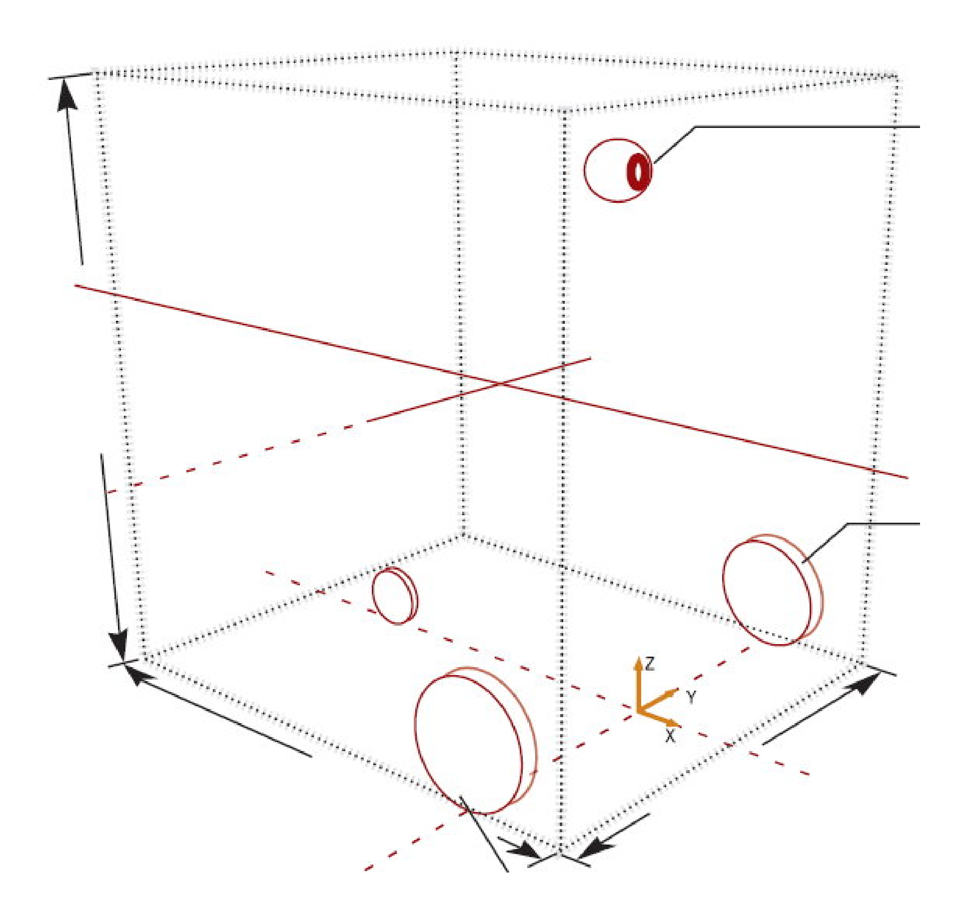
Result	Description
RESULT_BUSY	The StarGazer is busy, please try it later.
RESULT_SYSTEM_ERROR	System error.
RESULT_IO_ERROR	Local input/output error.
RESULT_SUCCESS	The function is successfully implemented.



Resource Configuration File Setup

The configuration files include Comm, IR, Sonar, Bumper, Driver, and Power. Users could determine the parameters in *.conf based on ones hardware platform. If any hardware is changed, for example, sensors or motors, users are only needed to make the related modification of *.conf.

The following picture shows the origin and coordinate definition. All the positions of sensors and drive system are based on the defined coordinate system, and the unit is mm.



Comm.conf

Name:Comm .... Object name

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



Number:1 .... Object quantity

Count:1... Object number
CommPort:COM2... COM port
BaudRate:115200... Baud rate
StopBit:0 .... Stop bit
ByteSize:3.. Data size

//
//CommPort
//COM1~COM10.....
//
//BaudRate
//57600
//115200
//...
//
//StopBit
//0 SB1
//1 SB1.5
//2 SB2
//
//ByteSize
//0 DB5
//1 DB6
//2 DB7
//3 DB8

Sonar.conf & IR.conf & Bumper.conf

Name:Sensor .... Object name
Number:2 .... Object quantity

Count:1... Object number
Manufacturer:SRF... Manufacturer
Model:SRF05... Object Model
Type:1... Object type



_x:0 .... Position <mm>
_y:207.415
_z:180
_roll:0.. Posture <deg>
_pitch:0
_yaw:90
MaxDistance:500 .... Max distance reading<mm>
MinDistance:10... Min distance reading<mm>
Radius:0.32 .... Radius<deg>

Count:2
Manufacturer:SRF
Model:SRF05
Type:1
_x:167.802
_y:121.915
_z:180
_roll:0
_pitch:0
_yaw:36
MaxDistance:500
MinDistance:10
Radius:0.32

//
//Type
//0=IR_Sensor
//1=Sonar_Sensor
//2=Camera_Sensor
//

Drive.conf
Name:Drive.. Object name
Number:1 .... Object quantity

Count:Left... Object number

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



MotorManufacturer:maxon.. Motor Manufacturer
MotorModel:MaxonMotor.. Motor Model
MotorType:0.. Motor Type
MotorStallTorque:.. Max stall torque <g-mm>
MotorMaxSpeed:6000 .... Max speed <RPM>
GearBoxManufacturer:maxon .... Gear Manufacturer
GearBoxModel:MaxonGear.. Gear Model
GearBoxType:0 .... Gear Type
GearBoxStage:1 .... Gear Speed Transmission
GearBoxGearRatio:51... Gear Ratio
EncoderManufacturer:maxon... Encoder Manufacturer
EncoderModel:MaxonEncoder .... Encoder Model
EncoderType:0 .... Encoder Type
EncoderMaxBoundsAngle:0... Max Encoder Range <deg>
EncoderMinBoundsAngle:0.. Min Encoder Range <deg>
EncoderValuePerCircle:1024 .... Count per revolution
TireManufacturer:Toy .... Wheel Manufacturer
TireModel:ToyTire .... Wheel Model
TireWidth:25 .... Wheel width
TireCircumference:660 .... Wheel circumference
TireRadius:50.. Wheel radius
MotorDriverManufacturer:MSI .... Motor driver manufacturer
MotorDriverModel:Ver0.0 .... Motor driver model
MotorDriverType:0 .... Driver type
MotorDriverMaxOutputValue:3000... Max driver output energy
MotorDriverMinOutputValue:-3000 .... Min driver output energy
_x:0 .... Wheel position <mm>
_y:162
_z:50
_roll:0.. Wheel posture <deg>
_pitch:0
_yaw:0

//
// MotorType
//0= DC_Blush
//1= DC_NonBlush

MICRO-STAR INT'L CO., LTD.
No.69,Li-De St.,Jung-He City,Taipei Hsien,Taiwan
TEL: 886-2-3234-5599 (REP)
FAX: 886-2-3234-5488 (REP)



//2= AC_Blush
//3= AC_NonBlush

//
// EncoderType
//0= Optics
//1= Resistance
//

//
// GearBoxType
//0= Fix
//1= MT
//2= CVT
//

//
// MotorDriverType
//0= TwoPhaseHBridge
//1= ThreePhaseHBridge
//

Power.conf
Name:Power.. Object name
Number:1 .... Object quantity

Count:1.. Object number
BatteryManufacturer:Pd.. Battery Manufacturer
BatteryModel:Pd .... Battery Model
BatteryMaxCurrent:0 .... Max current <A/hr>
BatteryMaxVoltage:24 .... Max voltage <V>
BatteryMinVoltage:20... Min voltage <V>

ChargerManufacturer:MSI... Charger manufacturer
ChargerModel:Ver0.0.. Charger model
ChargerMaxCurrent:2 .... Max charger current<A/hr>
ChargerMaxVoltage:26 .... Max charger voltage <V>